Hospital Management System Project Report

Team Details :

Tanay Balakrishna

Adithya KR

1.About:

The proposed Hospital Management System (HMS) is a comprehensive software solution designed to streamline hospital operations, including patient registration, appointment scheduling, medical history tracking, billing, and staff management. Built on a centralized MySQL database, the system ensures secure and efficient handling of patient and administrative data. With a user-friendly interface, the HMS facilitates seamless interaction for healthcare providers, administrators, and patients, enhancing resource utilization. Scalable and compliant with healthcare regulations, this system promotes operational efficiency, data integrity, and improved patient care.

2. User requirement:

2.1 User Management

- **Admins can manage user roles, profiles, and authentication details.**

2.2 Asset Management

- **Track and monitor hospital assets, including vendors and departments.**

2.3 Patient Record Management

- **Store and access detailed patient information such as personal details, ailments, and discharge status.**

2.4 Medical Inventory Management

- **Manage pharmaceuticals and medical equipment, including vendors and stock levels.**

2.5 Payroll Automation

- Generate and manage payrolls for hospital staff with salary details and payment status.

2.6 Laboratory Operations

- Record and retrieve laboratory test results for patients.

2.7 Doctor Profiles and Surgery Management

- Maintain doctor information and log surgery details, including patient and procedure data.

2.8 Financial Management

- Manage accounts, transactions, and financial records for the hospital.

2.9 Prescription Management

- Generate and maintain prescriptions for patients, including instructions and ailment details.

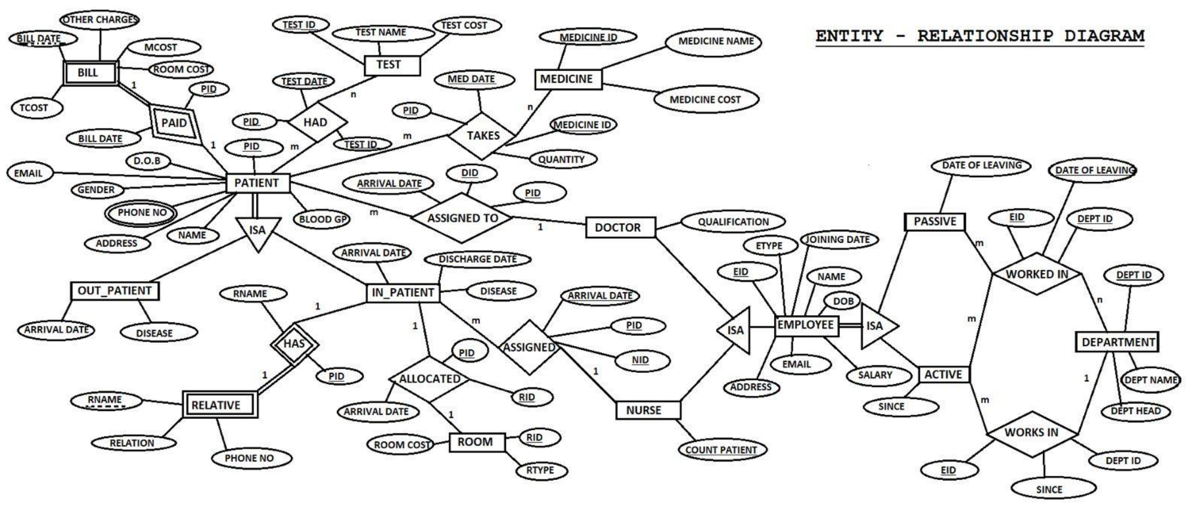
2.10 Surgery and Transfers

- Log patient transfers and surgery-related records for better tracking

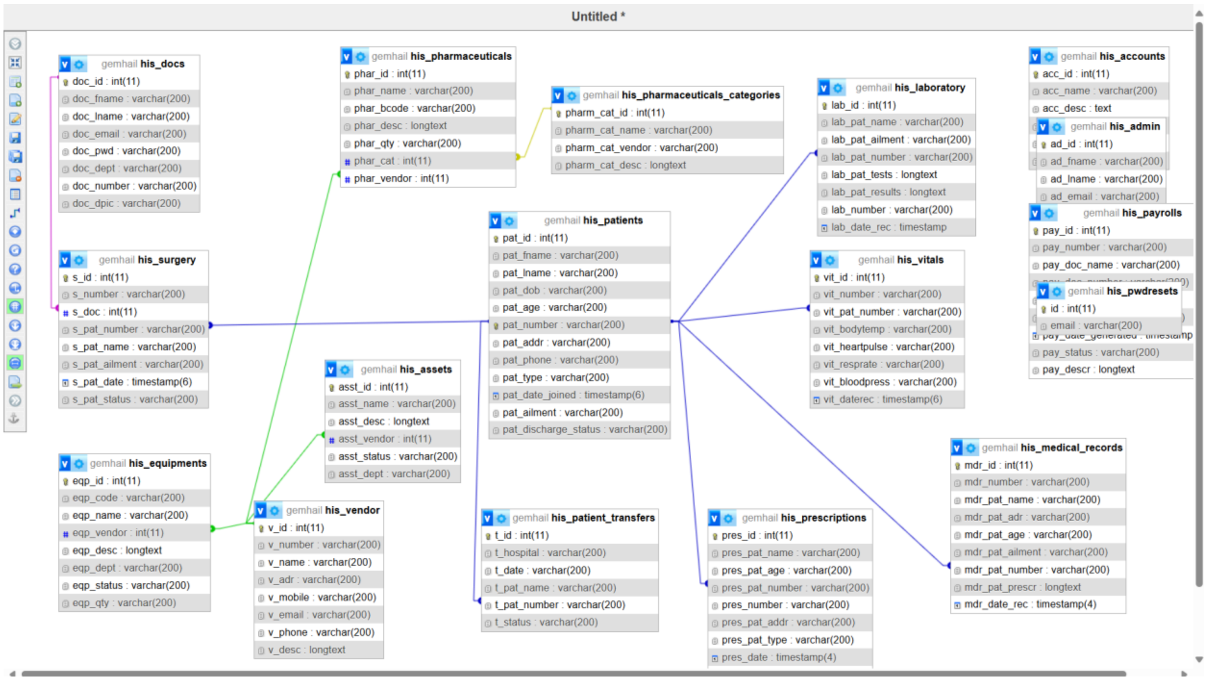
3. Software/Tools/Programming languages:

- PHP
- MySQL (Database)
- Phpmyadmin (Accessing the database)
- CSS, JavaScript and SASS

4. ER Diagram:

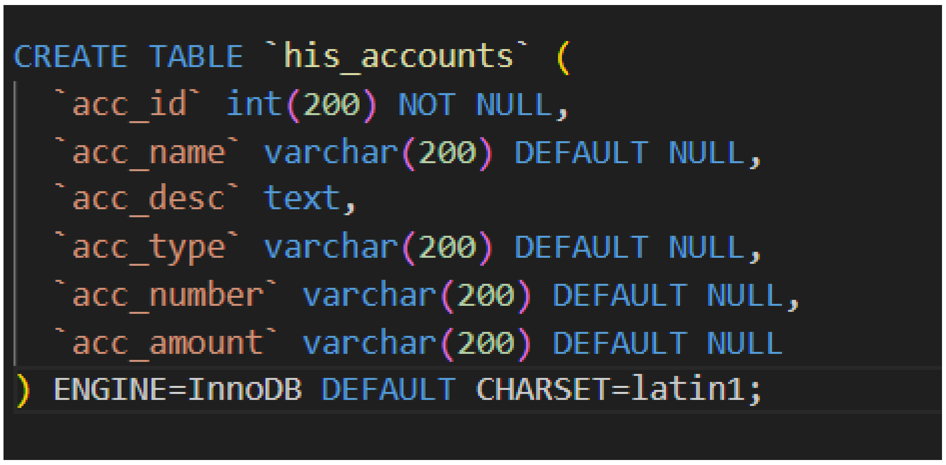


5. Relational Schema:

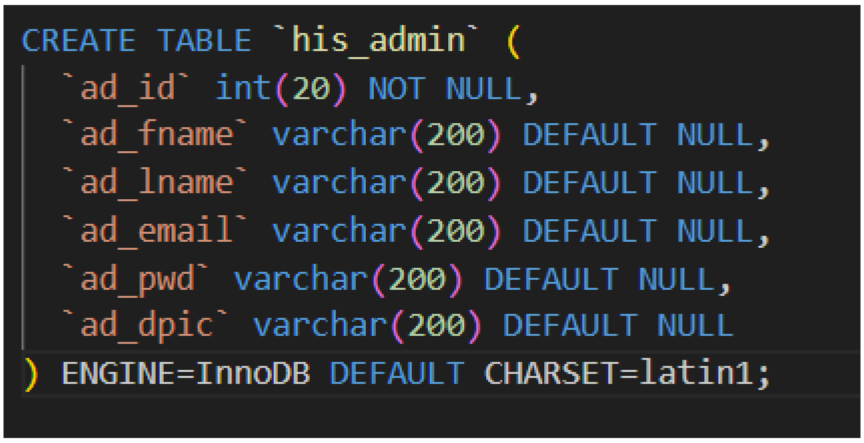


6.DDL Commands:

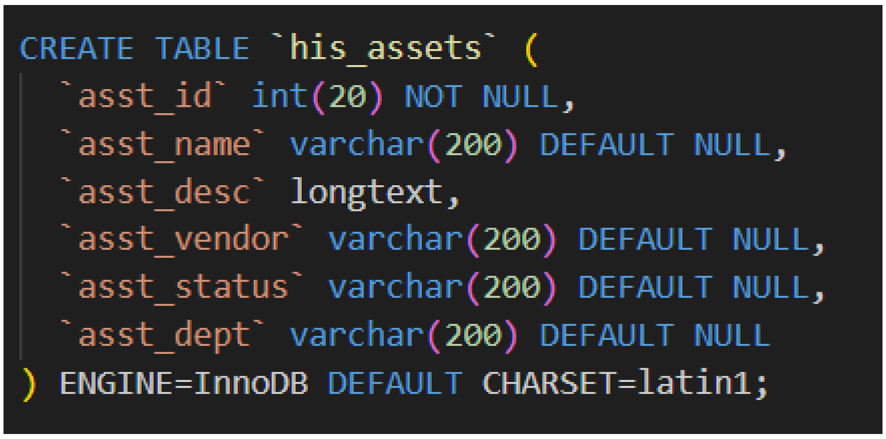
These are the create table commands:



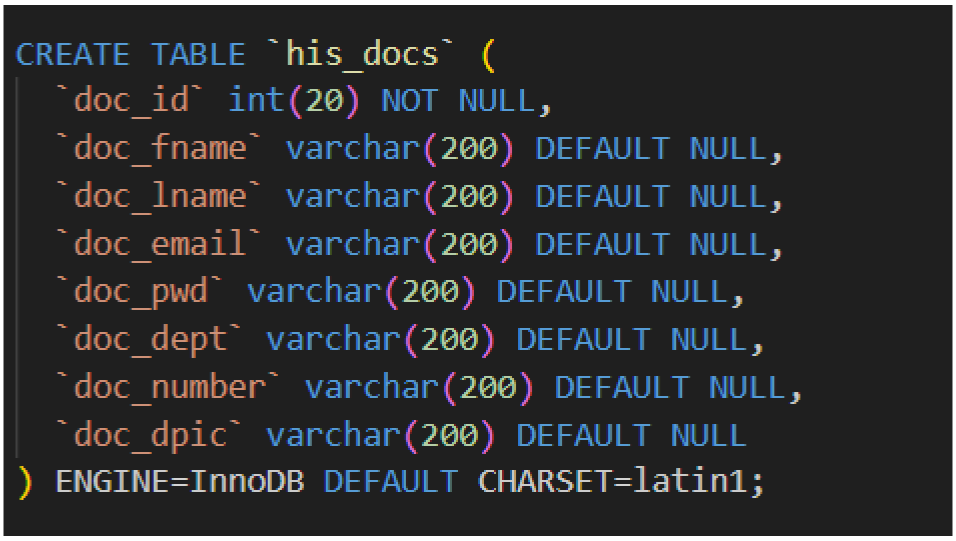
```
CREATE TABLE `his_accounts` (
  `acc_id` int(200) NOT NULL,
  `acc_name` varchar(200) DEFAULT NULL,
  `acc_desc` text,
  `acc_type` varchar(200) DEFAULT NULL,
  `acc_number` varchar(200) DEFAULT NULL,
  `acc_amount` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



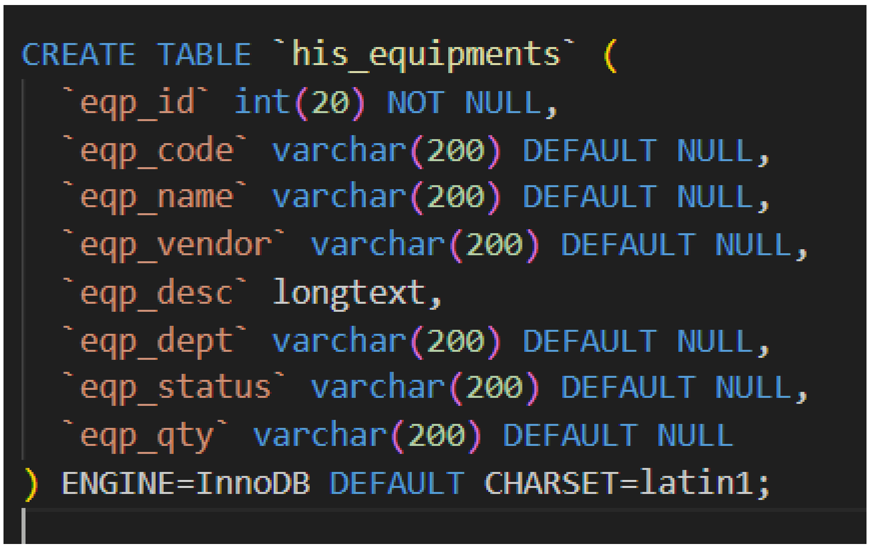
```sql
CREATE TABLE `his_admin` (
  `ad_id` int(20) NOT NULL,
  `ad_fname` varchar(200) DEFAULT NULL,
  `ad_lname` varchar(200) DEFAULT NULL,
  `ad_email` varchar(200) DEFAULT NULL,
  `ad_pwd` varchar(200) DEFAULT NULL,
  `ad_dpic` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



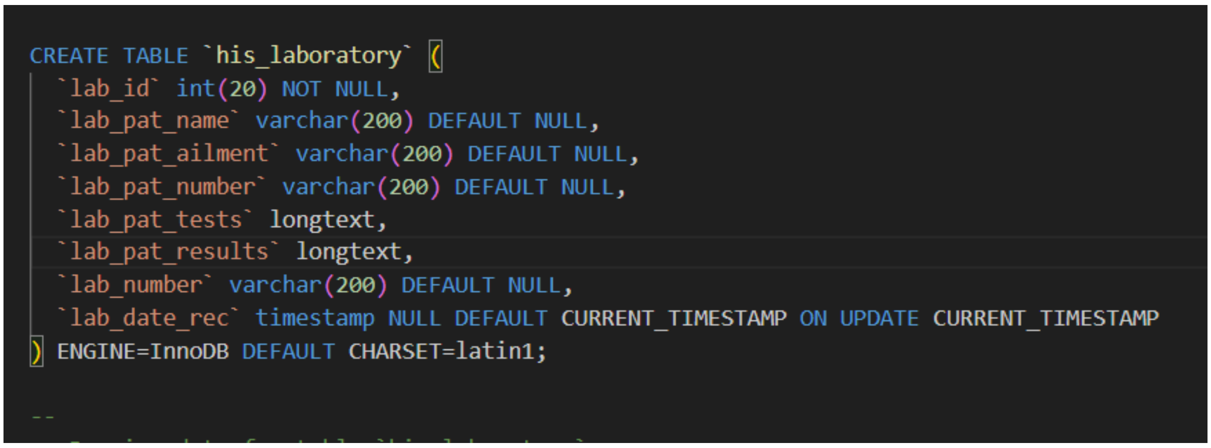
```sql
CREATE TABLE `his_assets` (
  `asst_id` int(20) NOT NULL,
  `asst_name` varchar(200) DEFAULT NULL,
  `asst_desc` longtext,
  `asst_vendor` varchar(200) DEFAULT NULL,
  `asst_status` varchar(200) DEFAULT NULL,
  `asst_dept` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



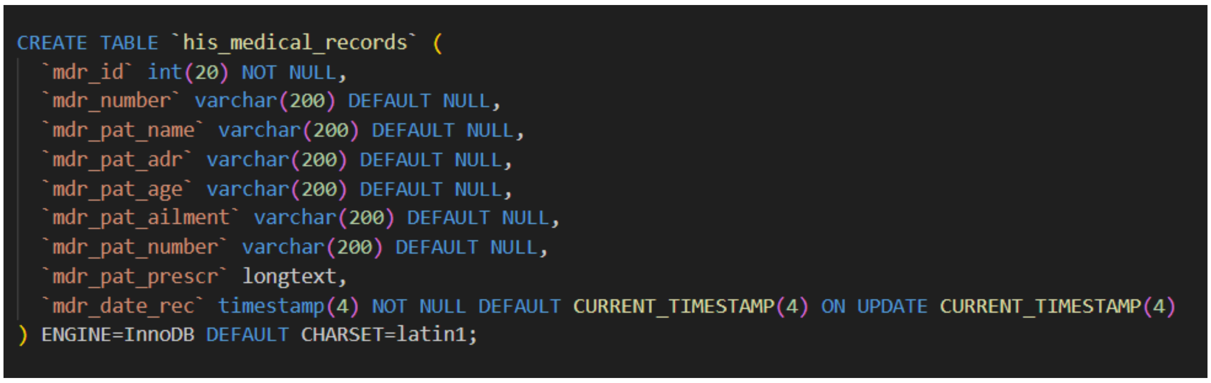
```sql
CREATE TABLE `his_docs` (
  `doc_id` int(20) NOT NULL,
  `doc_fname` varchar(200) DEFAULT NULL,
  `doc_lname` varchar(200) DEFAULT NULL,
  `doc_email` varchar(200) DEFAULT NULL,
  `doc_pwd` varchar(200) DEFAULT NULL,
  `doc_dept` varchar(200) DEFAULT NULL,
  `doc_number` varchar(200) DEFAULT NULL,
  `doc_dpic` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



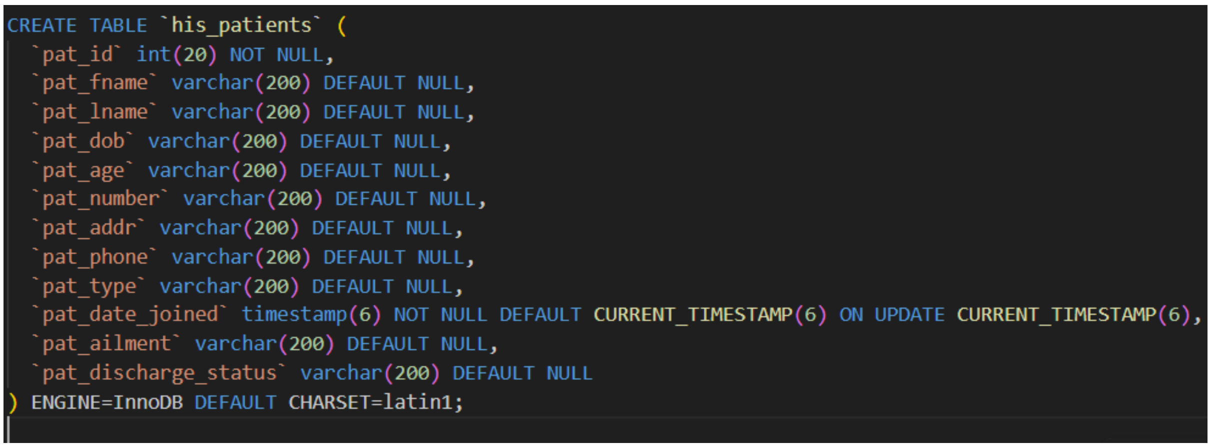
```sql
CREATE TABLE `his_equipments` (
  `eqp_id` int(20) NOT NULL,
  `eqp_code` varchar(200) DEFAULT NULL,
  `eqp_name` varchar(200) DEFAULT NULL,
  `eqp_vendor` varchar(200) DEFAULT NULL,
  `eqp_desc` longtext,
  `eqp_dept` varchar(200) DEFAULT NULL,
  `eqp_status` varchar(200) DEFAULT NULL,
  `eqp_qty` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



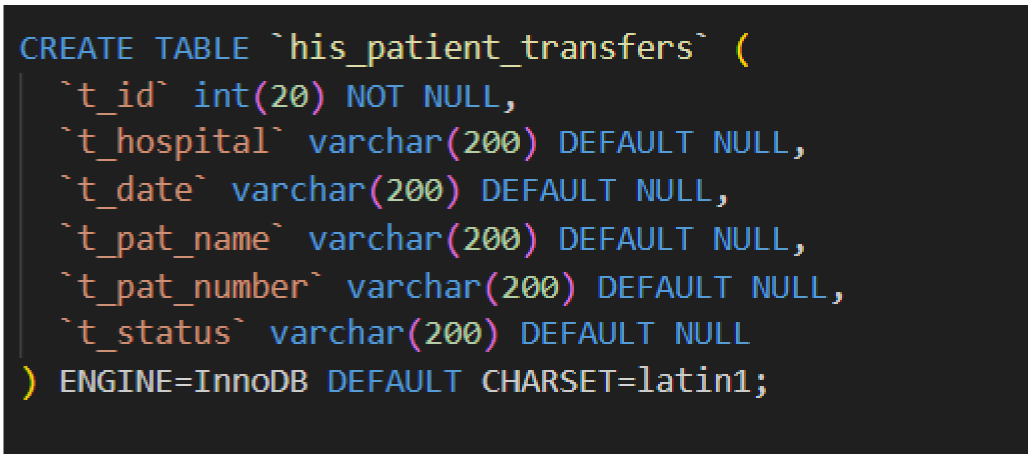
```sql
CREATE TABLE `his_laboratory` (
  `lab_id` int(20) NOT NULL,
  `lab_pat_name` varchar(200) DEFAULT NULL,
  `lab_pat_ailment` varchar(200) DEFAULT NULL,
  `lab_pat_number` varchar(200) DEFAULT NULL,
  `lab_pat_tests` longtext,
  `lab_pat_results` longtext,
  `lab_number` varchar(200) DEFAULT NULL,
  `lab_date_rec` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
```



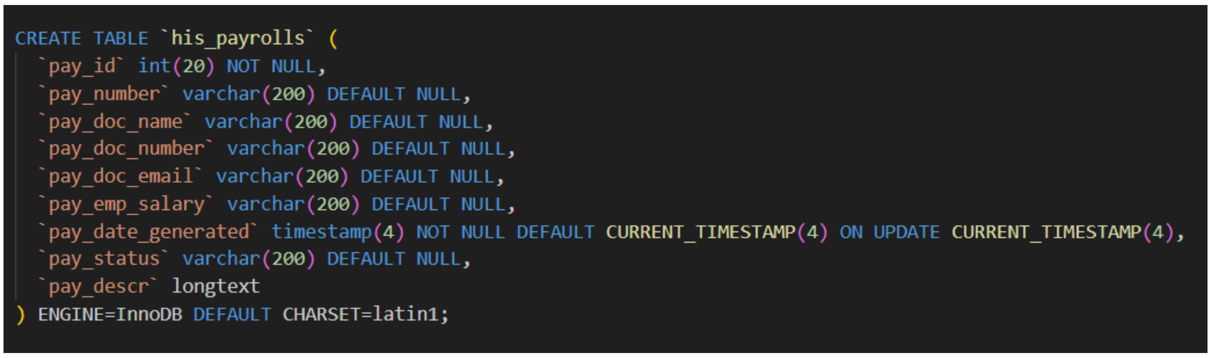
```sql
CREATE TABLE `his_medical_records` (
  `mdr_id` int(20) NOT NULL,
  `mdr_number` varchar(200) DEFAULT NULL,
  `mdr_pat_name` varchar(200) DEFAULT NULL,
  `mdr_pat_adr` varchar(200) DEFAULT NULL,
  `mdr_pat_age` varchar(200) DEFAULT NULL,
  `mdr_pat_ailment` varchar(200) DEFAULT NULL,
  `mdr_pat_number` varchar(200) DEFAULT NULL,
  `mdr_pat_prescr` longtext,
  `mdr_date_rec` timestamp(4) NOT NULL DEFAULT CURRENT_TIMESTAMP(4) ON UPDATE CURRENT_TIMESTAMP(4)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



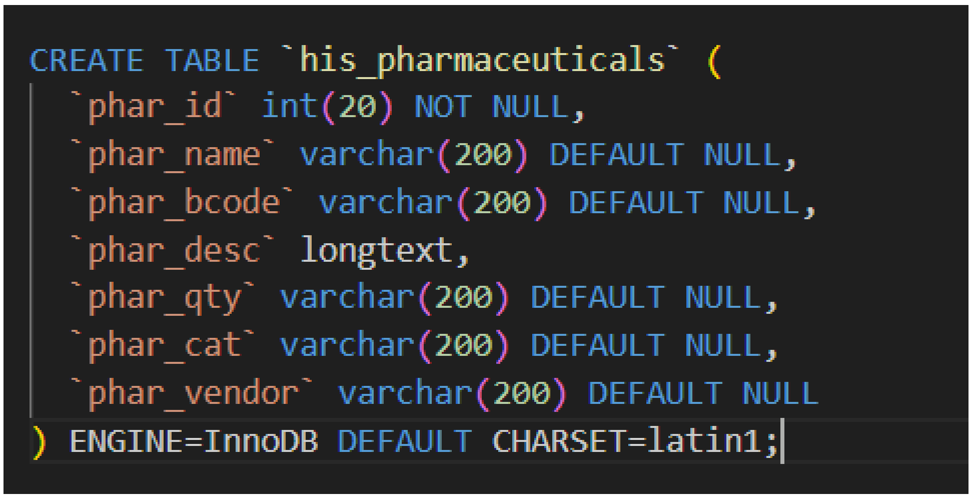
```sql
CREATE TABLE `his_patients` (
  `pat_id` int(20) NOT NULL,
  `pat_fname` varchar(200) DEFAULT NULL,
  `pat_lname` varchar(200) DEFAULT NULL,
  `pat_dob` varchar(200) DEFAULT NULL,
  `pat_age` varchar(200) DEFAULT NULL,
  `pat_number` varchar(200) DEFAULT NULL,
  `pat_addr` varchar(200) DEFAULT NULL,
  `pat_phone` varchar(200) DEFAULT NULL,
  `pat_type` varchar(200) DEFAULT NULL,
  `pat_date_joined` timestamp(6) NOT NULL DEFAULT CURRENT_TIMESTAMP(6) ON UPDATE CURRENT_TIMESTAMP(6),
  `pat_ailment` varchar(200) DEFAULT NULL,
  `pat_discharge_status` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



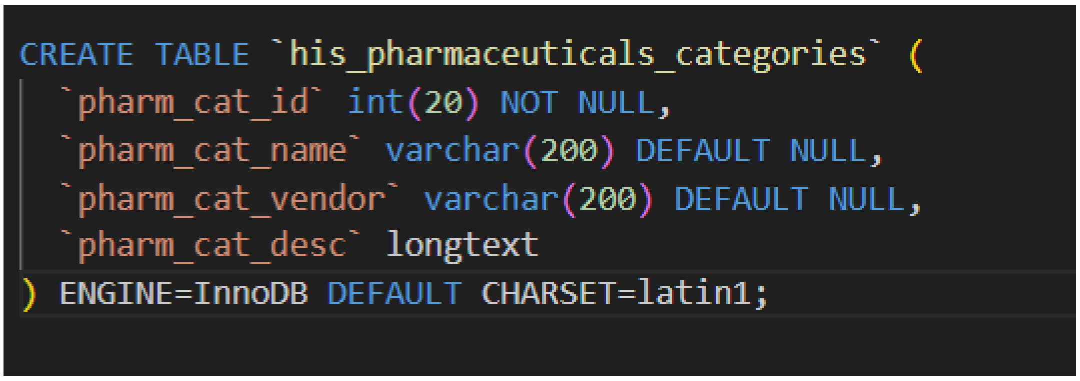
```sql
CREATE TABLE `his_patient_transfers` (
  `t_id` int(20) NOT NULL,
  `t_hospital` varchar(200) DEFAULT NULL,
  `t_date` varchar(200) DEFAULT NULL,
  `t_pat_name` varchar(200) DEFAULT NULL,
  `t_pat_number` varchar(200) DEFAULT NULL,
  `t_status` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



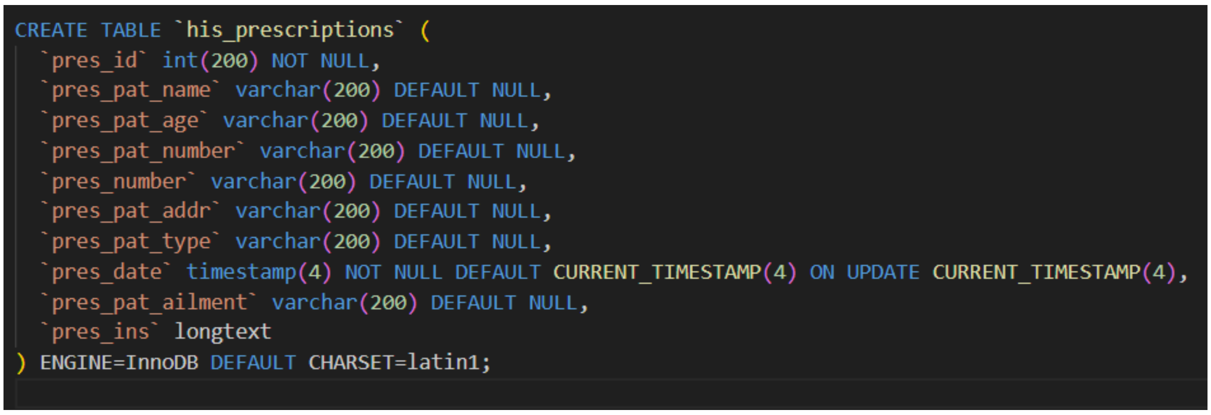
```sql
CREATE TABLE `his_payrolls` (
  `pay_id` int(20) NOT NULL,
  `pay_number` varchar(200) DEFAULT NULL,
  `pay_doc_name` varchar(200) DEFAULT NULL,
  `pay_doc_number` varchar(200) DEFAULT NULL,
  `pay_doc_email` varchar(200) DEFAULT NULL,
  `pay_emp_salary` varchar(200) DEFAULT NULL,
  `pay_date_generated` timestamp(4) NOT NULL DEFAULT CURRENT_TIMESTAMP(4) ON UPDATE CURRENT_TIMESTAMP(4),
  `pay_status` varchar(200) DEFAULT NULL,
  `pay_descr` longtext
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



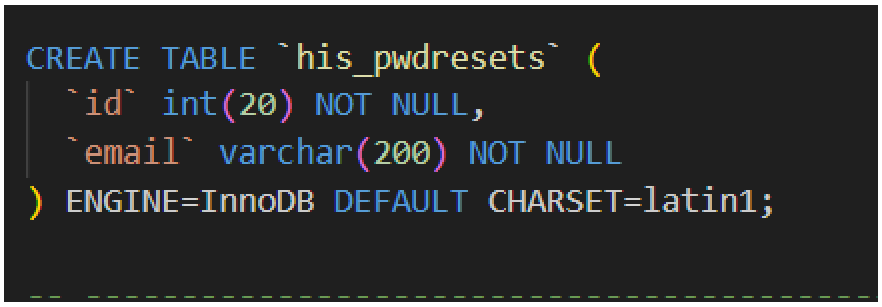
```sql
CREATE TABLE `his_pharmaceuticals` (
  `phar_id` int(20) NOT NULL,
  `phar_name` varchar(200) DEFAULT NULL,
  `phar_bcode` varchar(200) DEFAULT NULL,
  `phar_desc` longtext,
  `phar_qty` varchar(200) DEFAULT NULL,
  `phar_cat` varchar(200) DEFAULT NULL,
  `phar_vendor` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



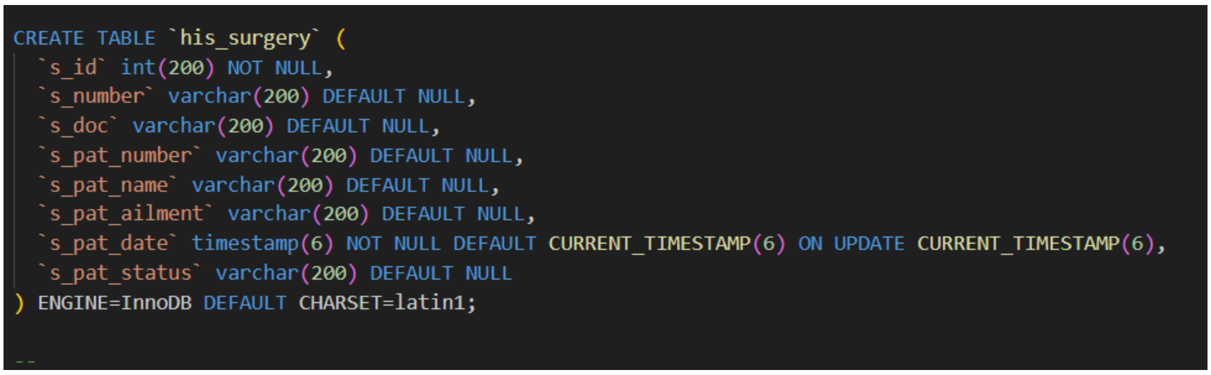
```sql
CREATE TABLE `his_pharmaceuticals_categories` (
  `pharm_cat_id` int(20) NOT NULL,
  `pharm_cat_name` varchar(200) DEFAULT NULL,
  `pharm_cat_vendor` varchar(200) DEFAULT NULL,
  `pharm_cat_desc` longtext
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



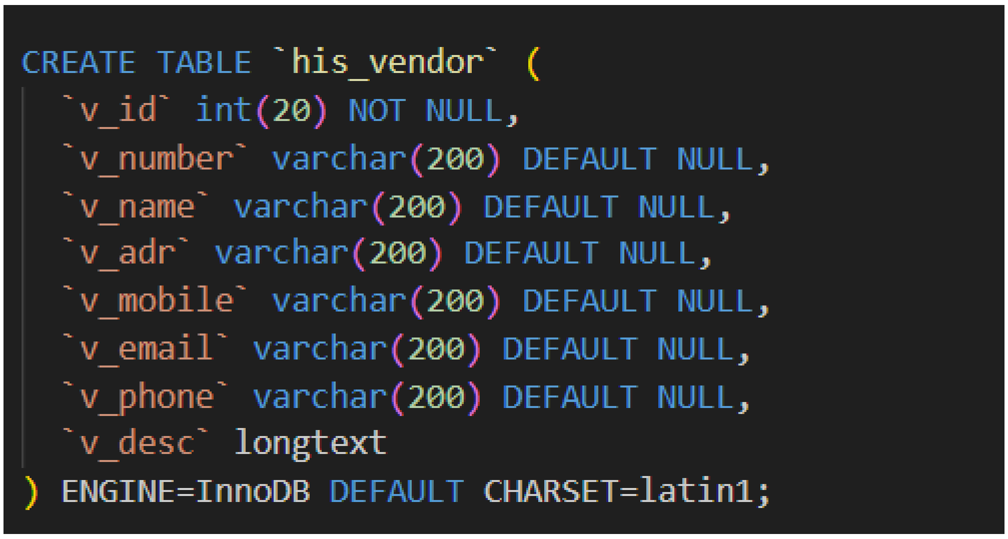
```sql
CREATE TABLE `his_prescriptions` (
  `pres_id` int(200) NOT NULL,
  `pres_pat_name` varchar(200) DEFAULT NULL,
  `pres_pat_age` varchar(200) DEFAULT NULL,
  `pres_pat_number` varchar(200) DEFAULT NULL,
  `pres_number` varchar(200) DEFAULT NULL,
  `pres_pat_addr` varchar(200) DEFAULT NULL,
  `pres_pat_type` varchar(200) DEFAULT NULL,
  `pres_date` timestamp(4) NOT NULL DEFAULT CURRENT_TIMESTAMP(4) ON UPDATE CURRENT_TIMESTAMP(4),
  `pres_pat_ailment` varchar(200) DEFAULT NULL,
  `pres_ins` longtext
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



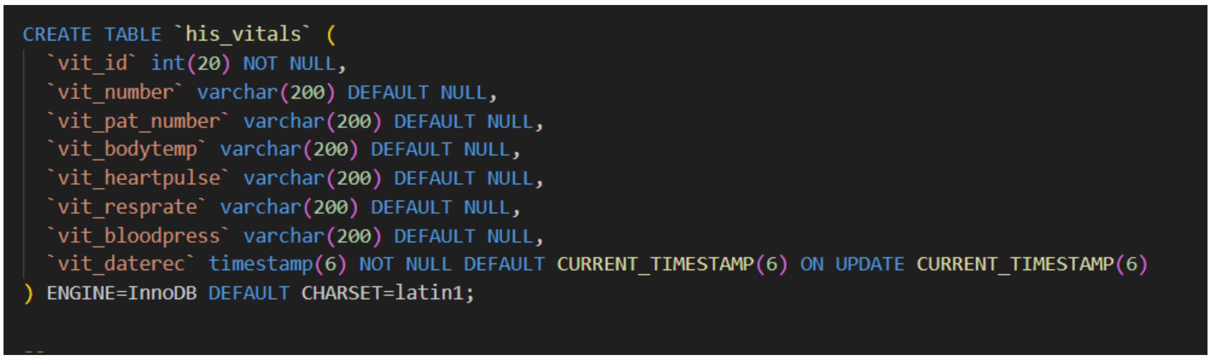
```sql
CREATE TABLE `his_pwdresets` (
  `id` int(20) NOT NULL,
  `email` varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


-- --------------------------------------------------------
```



```sql
CREATE TABLE `his_surgery` (
  `s_id` int(200) NOT NULL,
  `s_number` varchar(200) DEFAULT NULL,
  `s_doc` varchar(200) DEFAULT NULL,
  `s_pat_number` varchar(200) DEFAULT NULL,
  `s_pat_name` varchar(200) DEFAULT NULL,
  `s_pat_ailment` varchar(200) DEFAULT NULL,
  `s_pat_date` timestamp(6) NOT NULL DEFAULT CURRENT_TIMESTAMP(6) ON UPDATE CURRENT_TIMESTAMP(6),
  `s_pat_status` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
```



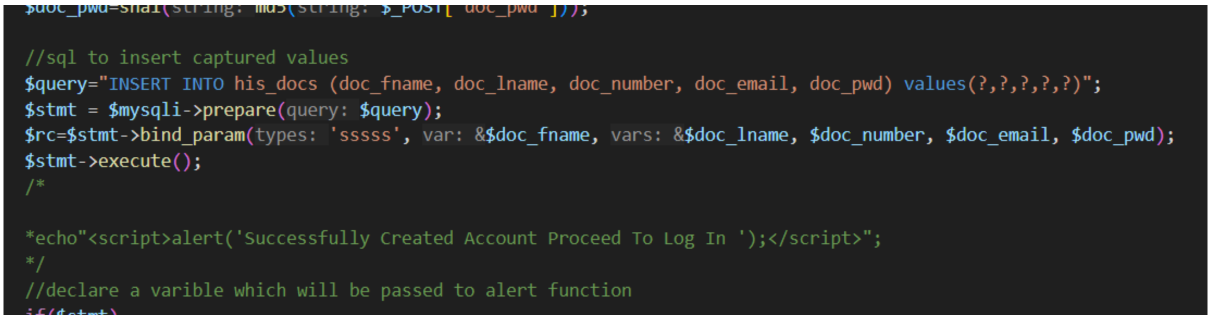
```sql
CREATE TABLE `his_vendor` (
  `v_id` int(20) NOT NULL,
  `v_number` varchar(200) DEFAULT NULL,
  `v_name` varchar(200) DEFAULT NULL,
  `v_adr` varchar(200) DEFAULT NULL,
  `v_mobile` varchar(200) DEFAULT NULL,
  `v_email` varchar(200) DEFAULT NULL,
  `v_phone` varchar(200) DEFAULT NULL,
  `v_desc` longtext
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



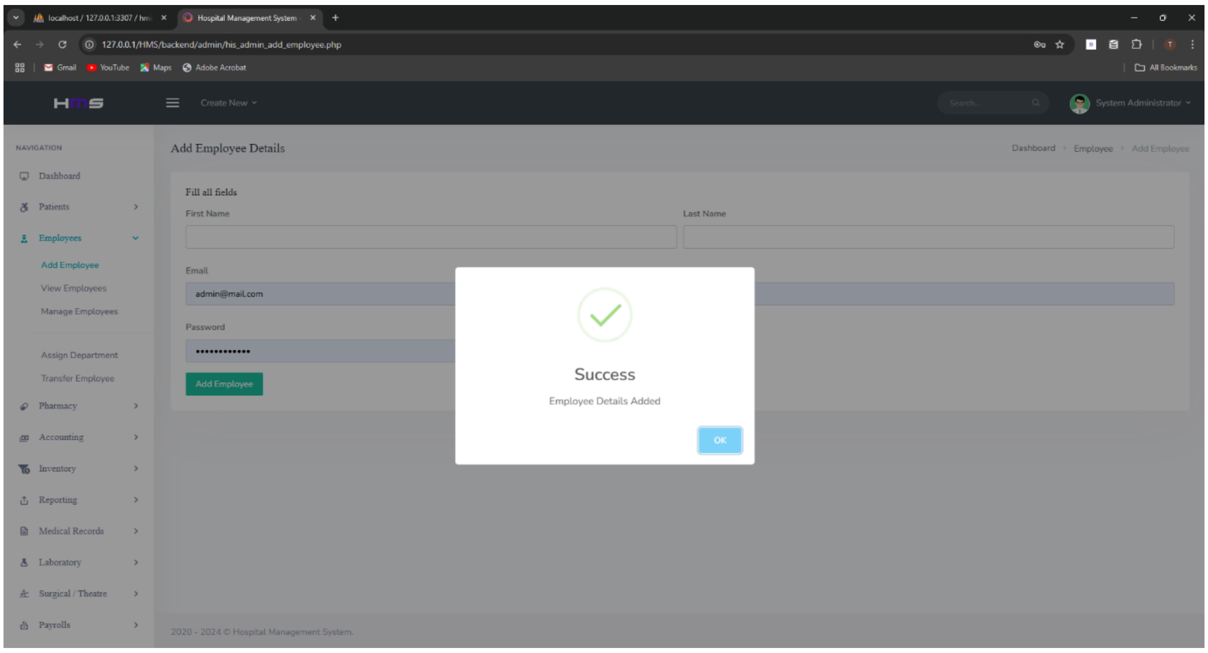
```sql
CREATE TABLE `his_vitals` (
  `vit_id` int(20) NOT NULL,
  `vit_number` varchar(200) DEFAULT NULL,
  `vit_pat_number` varchar(200) DEFAULT NULL,
  `vit_bodytemp` varchar(200) DEFAULT NULL,
  `vit_heartpulse` varchar(200) DEFAULT NULL,
  `vit_resprate` varchar(200) DEFAULT NULL,
  `vit_bloodpress` varchar(200) DEFAULT NULL,
  `vit_daterec` timestamp(6) NOT NULL DEFAULT CURRENT_TIMESTAMP(6) ON UPDATE CURRENT_TIMESTAMP(6)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
```

7.CRUD Operation:

7.1. Adding an employee (Jessica Smith)



```
$doc_pwd=sha1(string: md5(string: $_POST['doc_pwd']));

//sql to insert captured values
$query="INSERT INTO his_docs (doc_fname, doc_lname, doc_number, doc_email, doc_pwd) values(?,?,?,?,?)";
$stmt = $mysqli->prepare(query: $query);
$rc=$stmt->bind_param(types: 'sssss', var: &$doc_fname, vars: &$doc_lname, $doc_number, $doc_email, $doc_pwd);
$stmt->execute();
/*

*echo"<script>alert('Successfully Created Account Proceed To Log In ');</script>";
*/
//declare a varible which will be passed to alert function
if($stmt)
```



127.0.0.1/HMS/backend/admin/his_admin_add_employee.php

Add Employee Details

Fill all fields

First Name

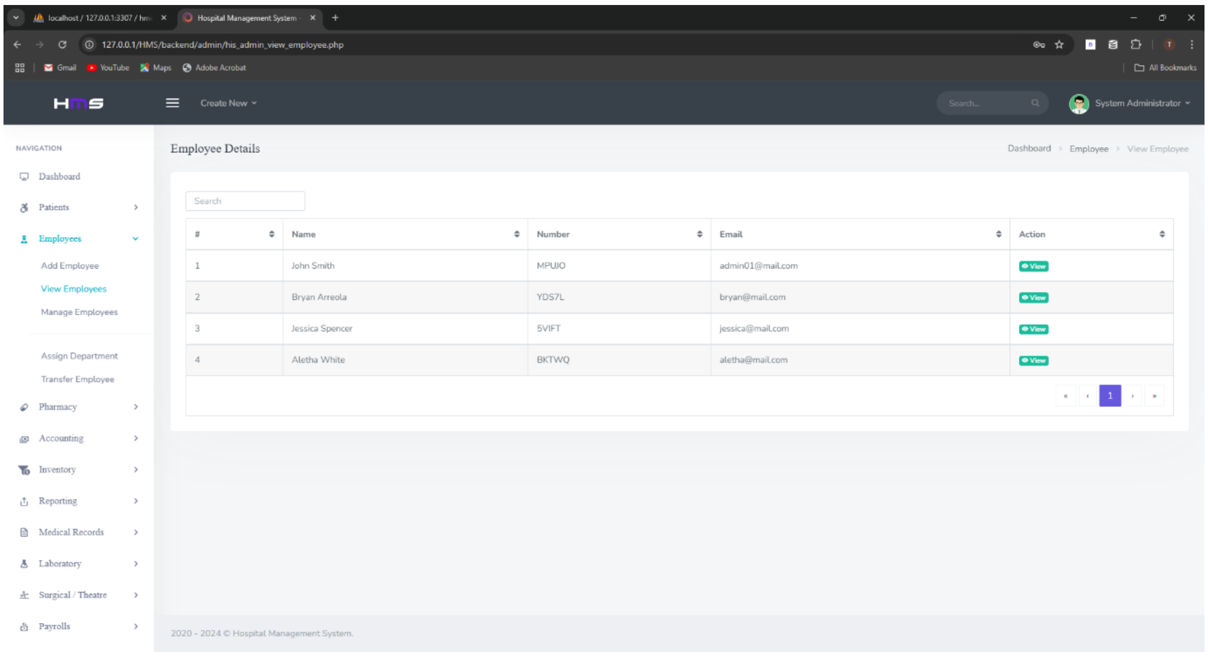
Last Name

Email: admin@mail.com

Password: ••••••••••••

Success

Employee Details Added

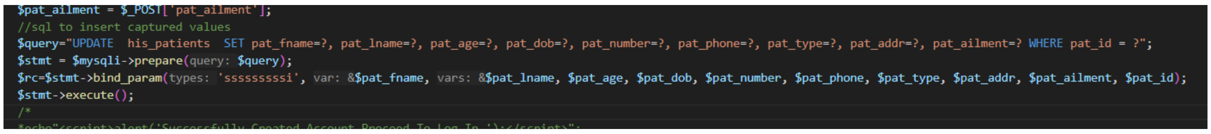


127.0.0.1/HMS/backend/admin/his_admin_view_employee.php

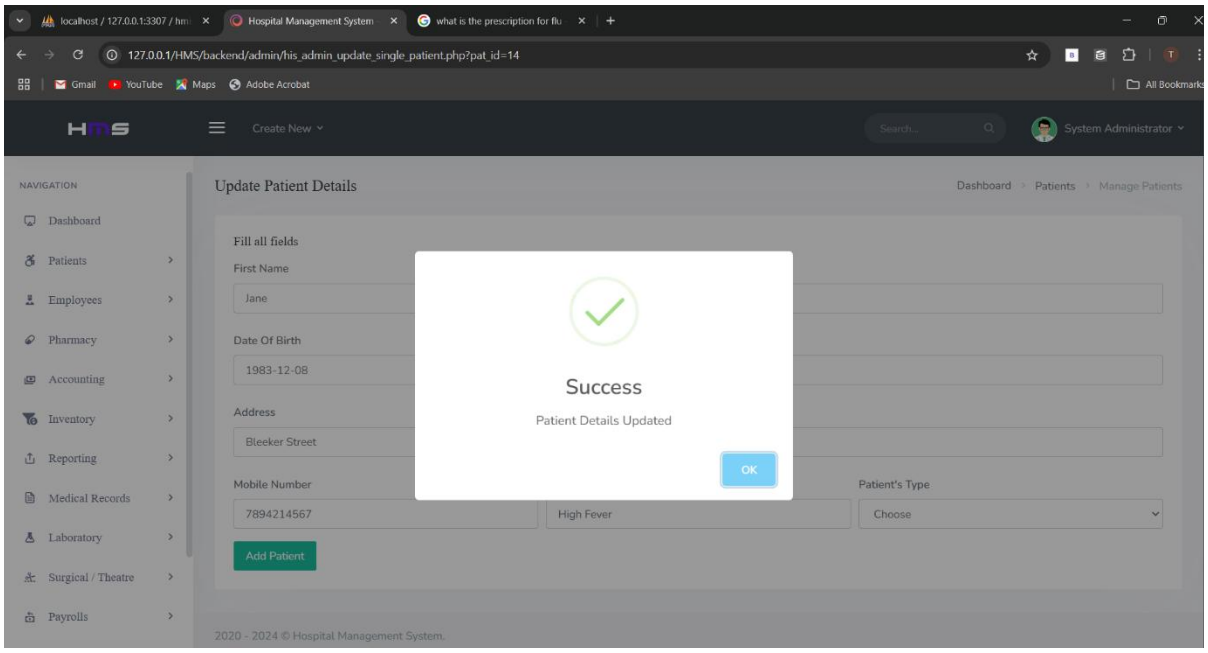
Employee Details

#	Name	Number	Email	Action
1	John Smith	MPUJO	admin01@mail.com	View
2	Bryan Arreola	YDS7L	bryan@mail.com	View
3	Jessica Spencer	5VIFT	jessica@mail.com	View
4	Aletha White	BKTWQ	aletha@mail.com	View

7.2.Updating a patient details:



```php
$pat_ailment = $_POST['pat_ailment'];
//sql to insert captured values
$query="UPDATE  his_patients  SET pat_fname=?, pat_lname=?, pat_age=?, pat_dob=?, pat_number=?, pat_phone=?, pat_type=?, pat_addr=?, pat_ailment=? WHERE pat_id = ?";
$stmt = $mysqli->prepare(query: $query);
$rc=$stmt->bind_param(types: 'sssssssssi', var: &$pat_fname, vars: &$pat_lname, $pat_age, $pat_dob, $pat_number, $pat_phone, $pat_type, $pat_addr, $pat_ailment, $pat_id);
$stmt->execute();
/*
*echo"<script>alert('Successfully Created Account Proceed To Log In ');</script>";
```



Update Patient Details

Dashboard > Patients > Manage Patients

Fill all fields

First Name: Jane

Date Of Birth: 1983-12-08

Address: Bleeker Street

Mobile Number: 7894214567

High Fever

Patient's Type: Choose

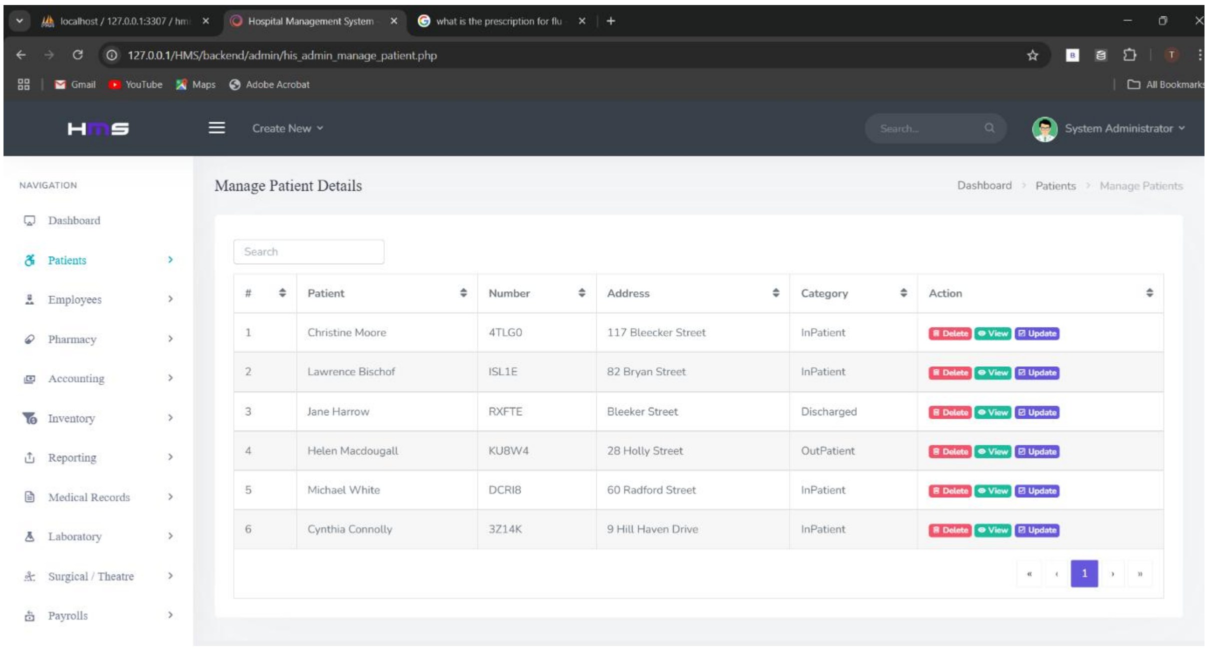
Add Patient

Success

Patient Details Updated

OK

2020 - 2024 © Hospital Management System.



Manage Patient Details

Dashboard > Patients > Manage Patients

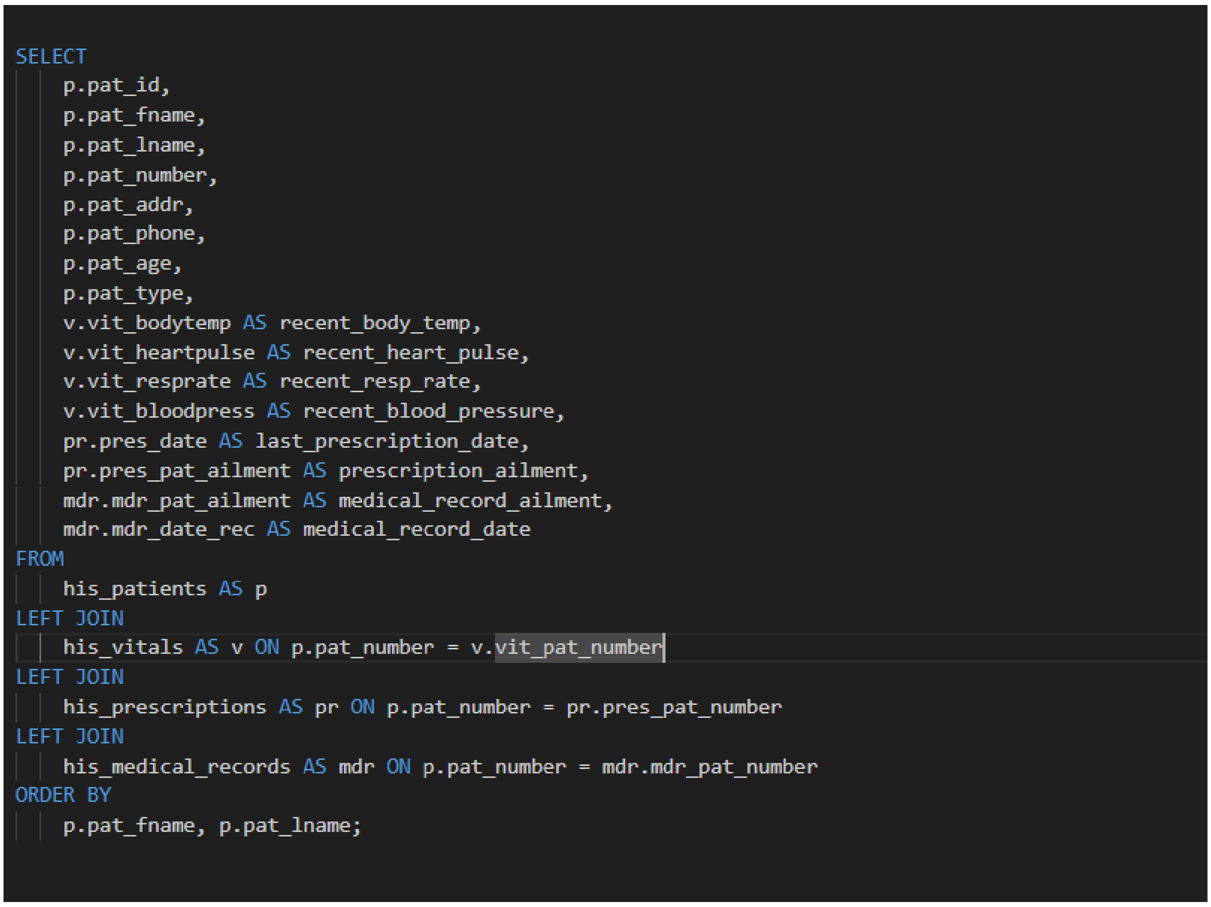
Search

#	Patient	Number	Address	Category	Action
1	Christine Moore	4TLG0	117 Bleecker Street	InPatient	Delete View Update
2	Lawrence Bischof	ISL1E	82 Bryan Street	InPatient	Delete View Update
3	Jane Harrow	RXFTE	Bleeker Street	Discharged	Delete View Update
4	Helen Macdougall	KU8W4	28 Holly Street	OutPatient	Delete View Update
5	Michael White	DCRI8	60 Radford Street	InPatient	Delete View Update
6	Cynthia Connolly	3Z14K	9 Hill Haven Drive	InPatient	Delete View Update

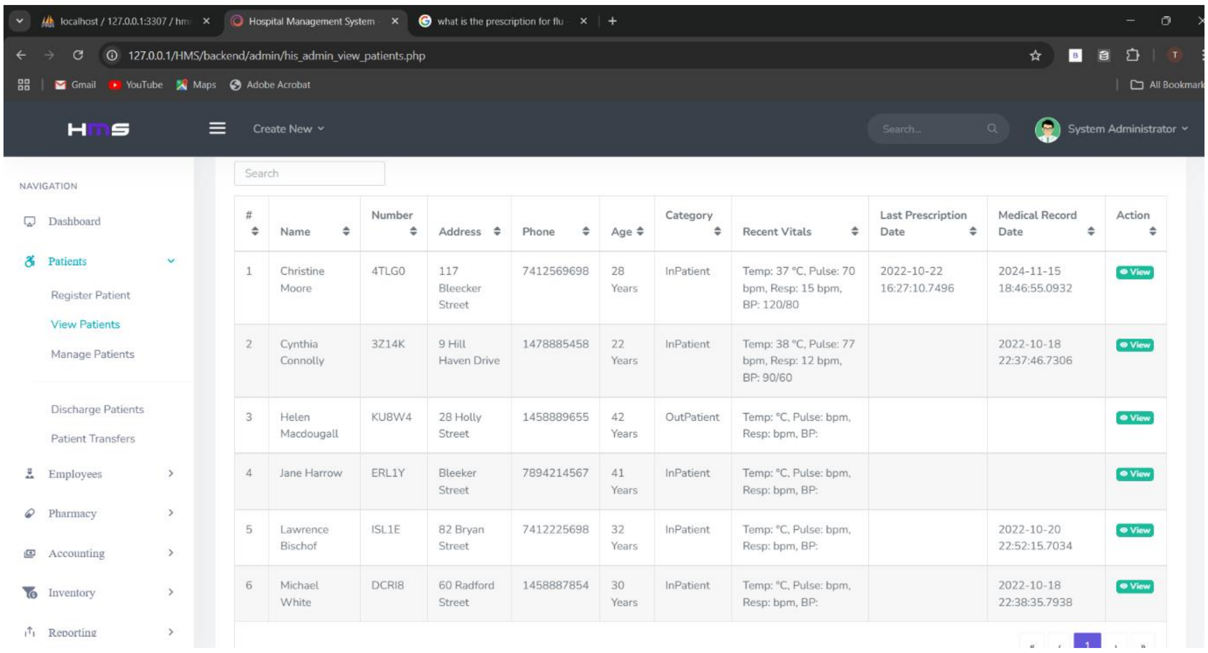
« ‹ 1 › »

8.Functionalities:

- **Join query (listing out all the details of the patient):**

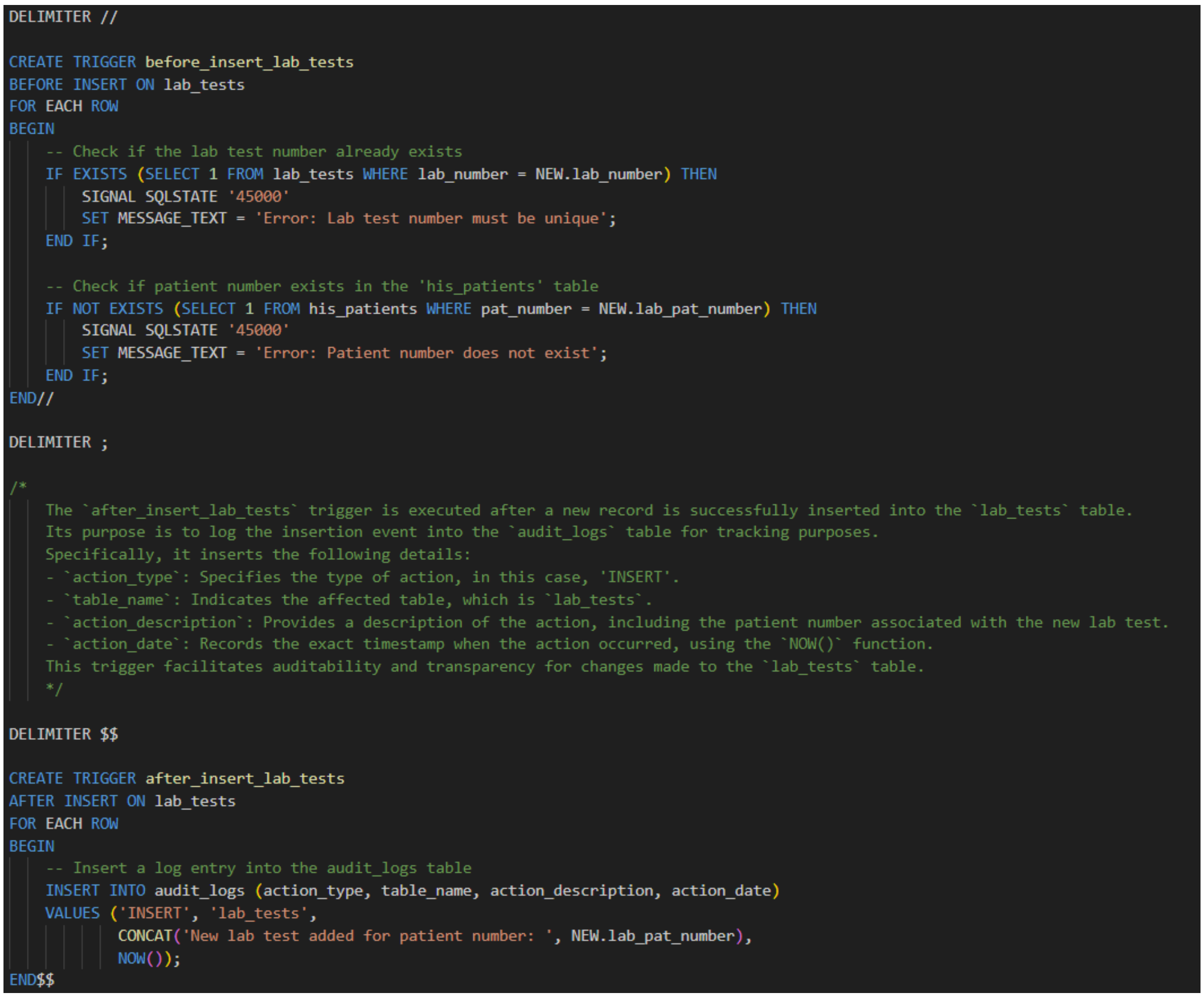


```
SELECT
    p.pat_id,
    p.pat_fname,
    p.pat_lname,
    p.pat_number,
    p.pat_addr,
    p.pat_phone,
    p.pat_age,
    p.pat_type,
    v.vit_bodytemp AS recent_body_temp,
    v.vit_heartpulse AS recent_heart_pulse,
    v.vit_resprate AS recent_resp_rate,
    v.vit_bloodpress AS recent_blood_pressure,
    pr.pres_date AS last_prescription_date,
    pr.pres_pat_ailment AS prescription_ailment,
    mdr.mdr_pat_ailment AS medical_record_ailment,
    mdr.mdr_date_rec AS medical_record_date
FROM
    his_patients AS p
LEFT JOIN
    his_vitals AS v ON p.pat_number = v.vit_pat_number
LEFT JOIN
    his_prescriptions AS pr ON p.pat_number = pr.pres_pat_number
LEFT JOIN
    his_medical_records AS mdr ON p.pat_number = mdr.mdr_pat_number
ORDER BY
    p.pat_fname, p.pat_lname;
```

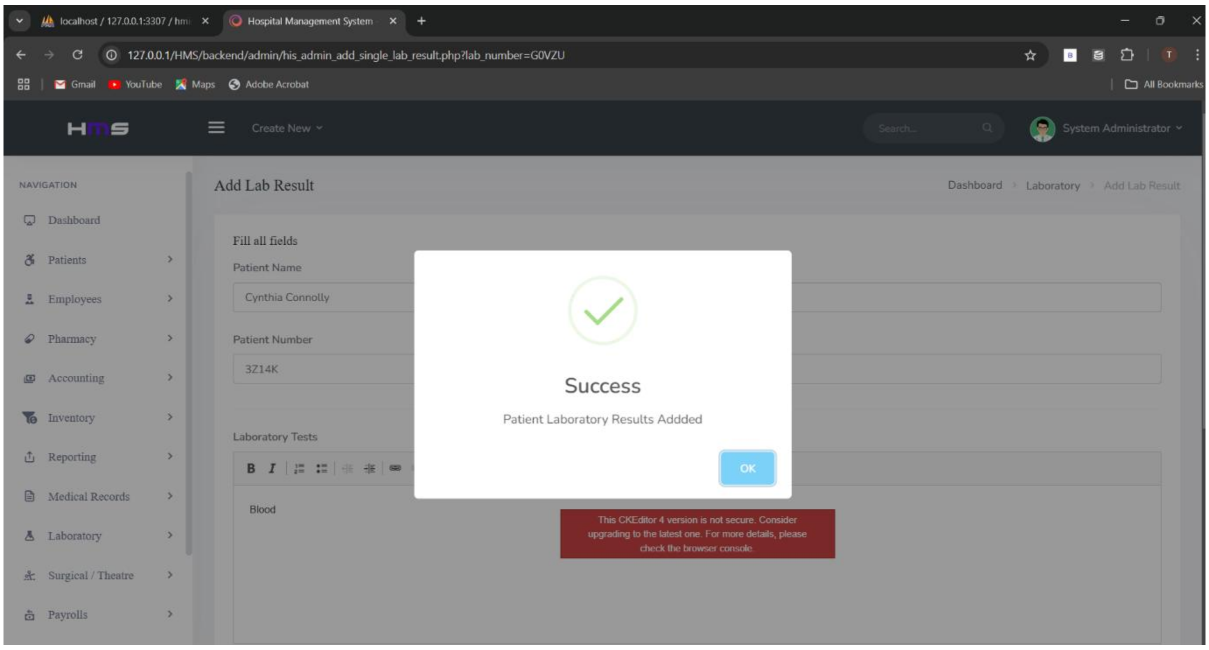


#	Name	Number	Address	Phone	Age	Category	Recent Vitals	Last Prescription Date	Medical Record Date	Action
1	Christine Moore	4TLG0	117 Bleecker Street	7412569698	28 Years	InPatient	Temp: 37 °C, Pulse: 70 bpm, Resp: 15 bpm, BP: 120/80	2022-10-22 16:27:10.7496	2024-11-15 18:46:55.0932	View
2	Cynthia Connolly	3Z14K	9 Hill Haven Drive	1478885458	22 Years	InPatient	Temp: 38 °C, Pulse: 77 bpm, Resp: 12 bpm, BP: 90/60		2022-10-18 22:37:46.7306	View
3	Helen Macdougall	KU8W4	28 Holly Street	1458889655	42 Years	OutPatient	Temp: °C, Pulse: bpm, Resp: bpm, BP:			View
4	Jane Harrow	ERL1Y	Bleeker Street	7894214567	41 Years	InPatient	Temp: °C, Pulse: bpm, Resp: bpm, BP:			View
5	Lawrence Bischof	ISL1E	82 Bryan Street	7412225698	32 Years	InPatient	Temp: °C, Pulse: bpm, Resp: bpm, BP:		2022-10-20 22:52:15.7034	View
6	Michael White	DCRI8	60 Radford Street	1458887854	30 Years	InPatient	Temp: °C, Pulse: bpm, Resp: bpm, BP:		2022-10-18 22:38:35.7938	View

- **Triggers:**



```sql
DELIMITER //

CREATE TRIGGER before_insert_lab_tests
BEFORE INSERT ON lab_tests
FOR EACH ROW
BEGIN
    -- Check if the lab test number already exists
    IF EXISTS (SELECT 1 FROM lab_tests WHERE lab_number = NEW.lab_number) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Error: Lab test number must be unique';
    END IF;

    -- Check if patient number exists in the 'his_patients' table
    IF NOT EXISTS (SELECT 1 FROM his_patients WHERE pat_number = NEW.lab_pat_number) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Error: Patient number does not exist';
    END IF;
END//

DELIMITER ;

/*
    The `after_insert_lab_tests` trigger is executed after a new record is successfully inserted into the `lab_tests` table.
    Its purpose is to log the insertion event into the `audit_logs` table for tracking purposes.
    Specifically, it inserts the following details:
    - `action_type`: Specifies the type of action, in this case, 'INSERT'.
    - `table_name`: Indicates the affected table, which is `lab_tests`.
    - `action_description`: Provides a description of the action, including the patient number associated with the new lab test.
    - `action_date`: Records the exact timestamp when the action occurred, using the `NOW()` function.
    This trigger facilitates auditability and transparency for changes made to the `lab_tests` table.
    */

DELIMITER $$

CREATE TRIGGER after_insert_lab_tests
AFTER INSERT ON lab_tests
FOR EACH ROW
BEGIN
    -- Insert a log entry into the audit_logs table
    INSERT INTO audit_logs (action_type, table_name, action_description, action_date)
    VALUES ('INSERT', 'lab_tests',
            CONCAT('New lab test added for patient number: ', NEW.lab_pat_number),
            NOW());
END$$
```



127.0.0.1/HMS/backend/admin/his_admin_add_single_lab_result.php?lab_number=G0VZU

Add Lab Result

Fill all fields

Patient Name: Cynthia Connolly

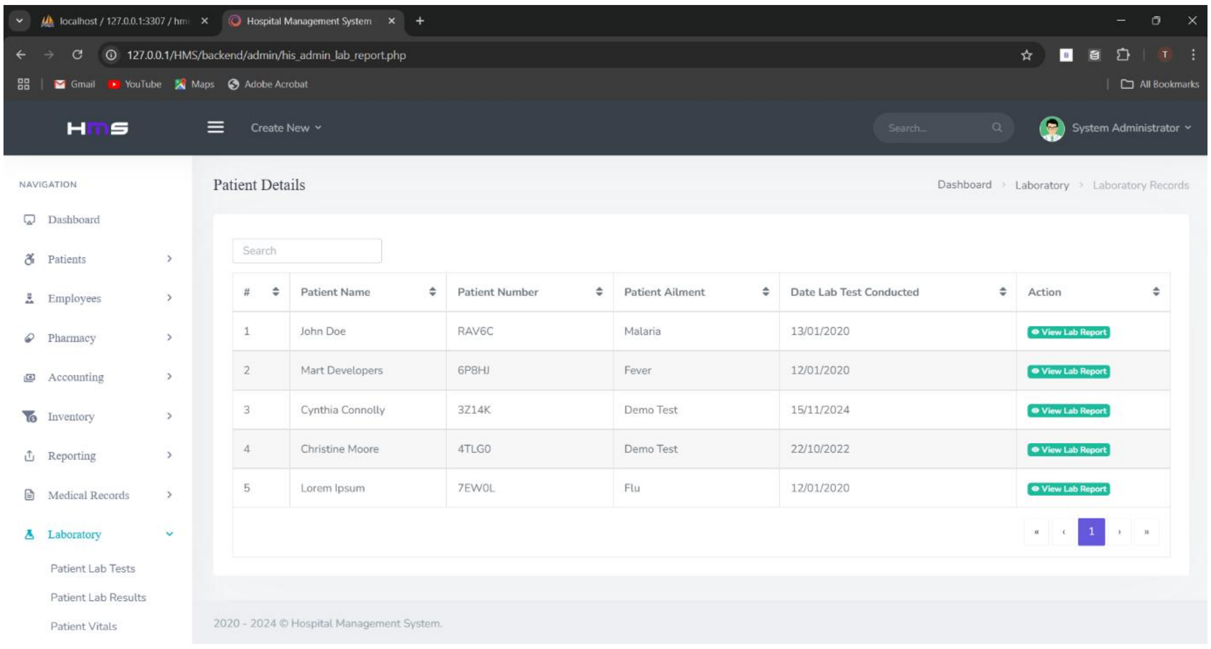
Patient Number: 3Z14K

Laboratory Tests: Blood

Success

Patient Laboratory Results Addded

This CKEditor 4 version is not secure. Consider upgrading to the latest one. For more details, please check the browser console.


localhost / 127.0.0.1:3307 / hm Hospital Management System

127.0.0.1/HMS/backend/admin/his_admin_lab_report.php

Gmail YouTube Maps Adobe Acrobat All Bookmarks

HMS Create New Search... System Administrator

NAVIGATION
- Dashboard
- Patients
- Employees
- Pharmacy
- Accounting
- Inventory
- Reporting
- Medical Records
- Laboratory
  - Patient Lab Tests
  - Patient Lab Results
  - Patient Vitals

Patient Details

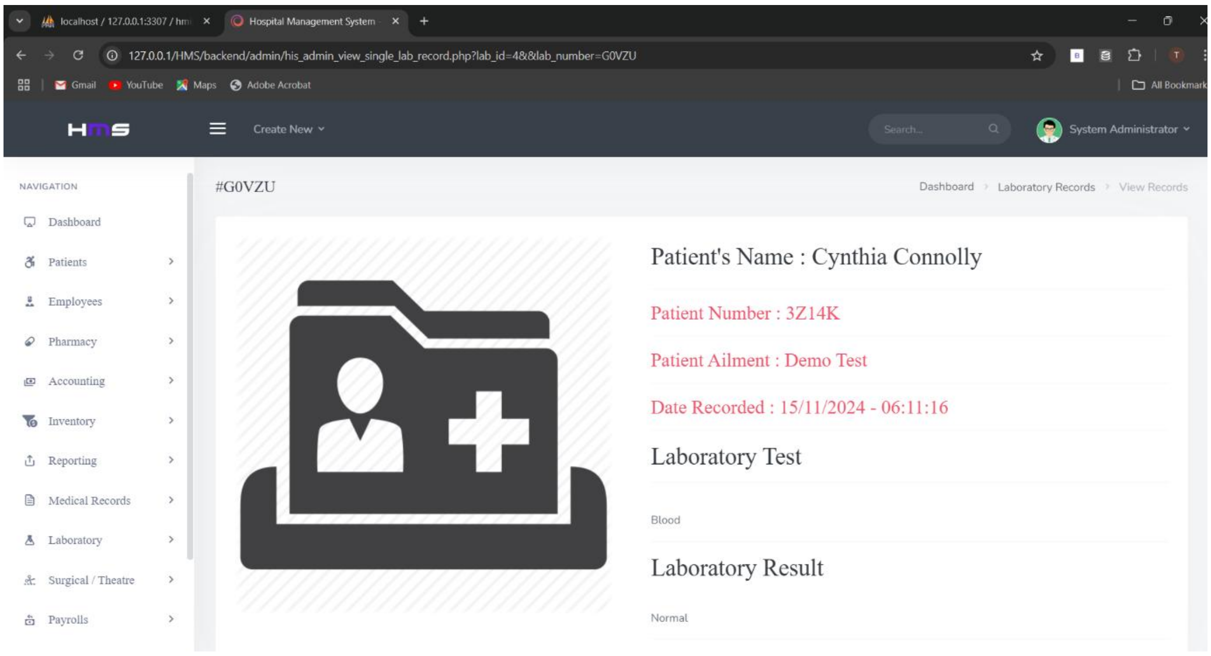
Dashboard > Laboratory > Laboratory Records

Search

#	Patient Name	Patient Number	Patient Ailment	Date Lab Test Conducted	Action
1	John Doe	RAV6C	Malaria	13/01/2020	View Lab Report
2	Mart Developers	6P8HJ	Fever	12/01/2020	View Lab Report
3	Cynthia Connolly	3Z14K	Demo Test	15/11/2024	View Lab Report
4	Christine Moore	4TLG0	Demo Test	22/10/2022	View Lab Report
5	Lorem Ipsum	7EW0L	Flu	12/01/2020	View Lab Report

1

2020 - 2024 © Hospital Management System.


localhost / 127.0.0.1:3307 / hm Hospital Management System

127.0.0.1/HMS/backend/admin/his_admin_view_single_lab_record.php?lab_id=4&&lab_number=G0VZU

Gmail YouTube Maps Adobe Acrobat All Bookmark

HMS Create New Search... System Administrator

NAVIGATION
- Dashboard
- Patients
- Employees
- Pharmacy
- Accounting
- Inventory
- Reporting
- Medical Records
- Laboratory
- Surgical / Theatre
- Payrolls

#G0VZU

Dashboard > Laboratory Records > View Records

Patient's Name : Cynthia Connolly

Patient Number : 3Z14K

Patient Ailment : Demo Test

Date Recorded : 15/11/2024 - 06:11:16

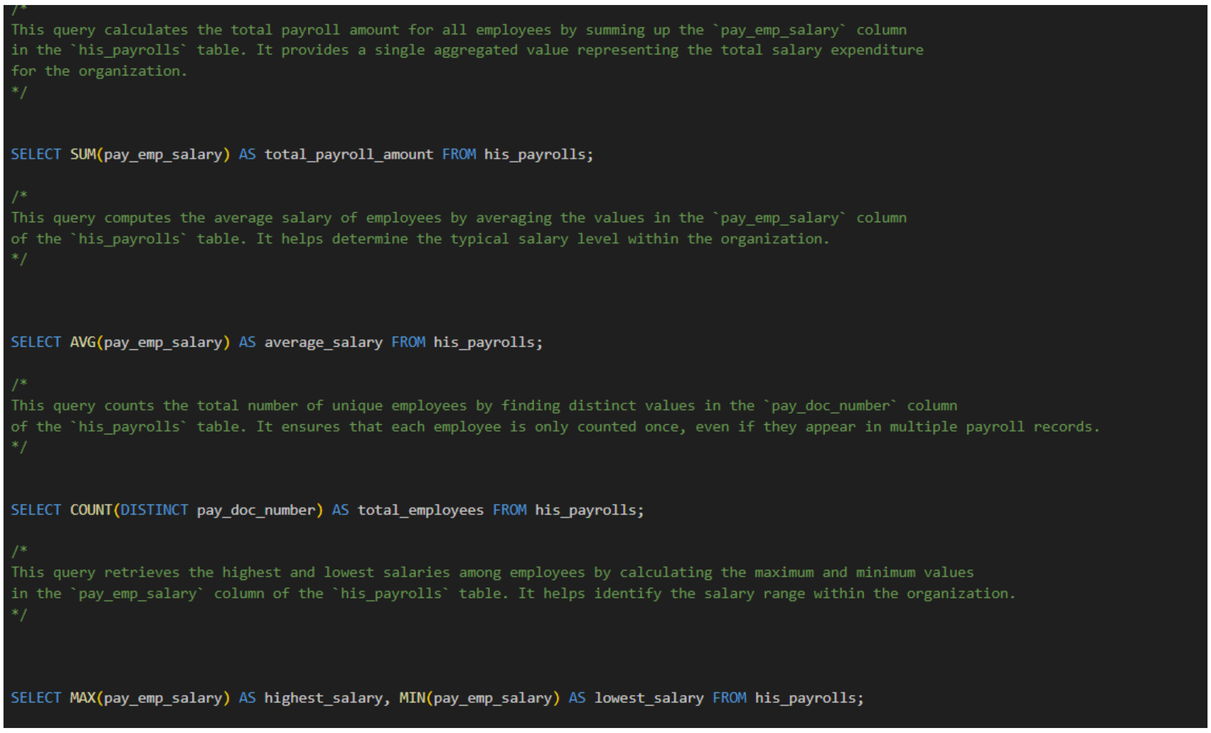
Laboratory Test

Blood

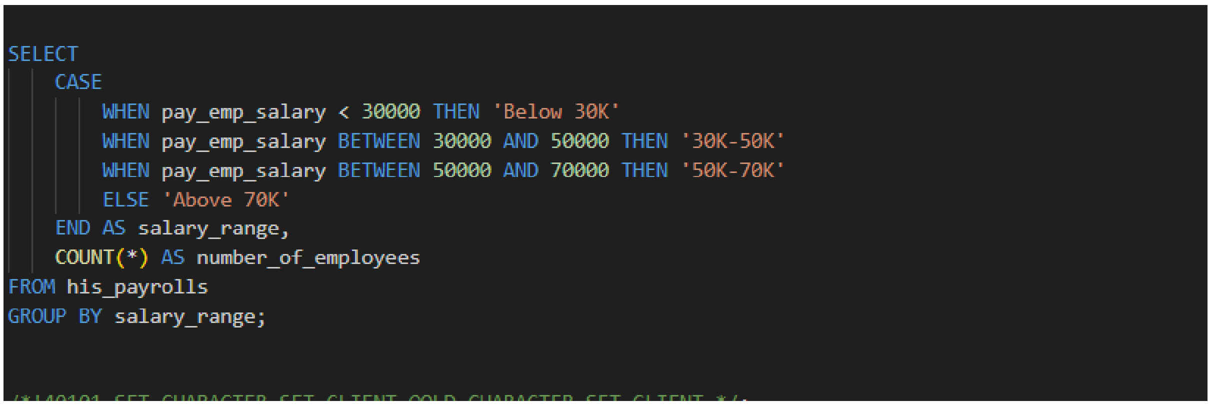
Laboratory Result

Normal

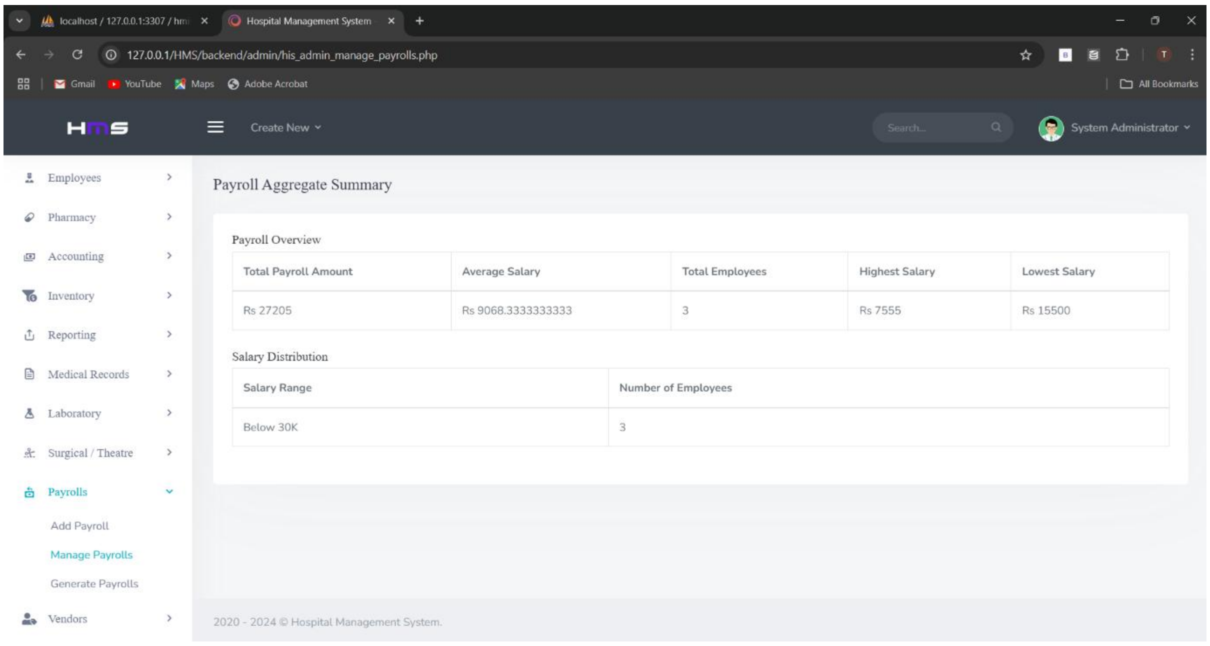
- **Aggregate function (used in payroll):**



```sql
/*
This query calculates the total payroll amount for all employees by summing up the `pay_emp_salary` column
in the `his_payrolls` table. It provides a single aggregated value representing the total salary expenditure
for the organization.
*/


SELECT SUM(pay_emp_salary) AS total_payroll_amount FROM his_payrolls;

/*
This query computes the average salary of employees by averaging the values in the `pay_emp_salary` column
of the `his_payrolls` table. It helps determine the typical salary level within the organization.
*/



SELECT AVG(pay_emp_salary) AS average_salary FROM his_payrolls;

/*
This query counts the total number of unique employees by finding distinct values in the `pay_doc_number` column
of the `his_payrolls` table. It ensures that each employee is only counted once, even if they appear in multiple payroll records.
*/


SELECT COUNT(DISTINCT pay_doc_number) AS total_employees FROM his_payrolls;

/*
This query retrieves the highest and lowest salaries among employees by calculating the maximum and minimum values
in the `pay_emp_salary` column of the `his_payrolls` table. It helps identify the salary range within the organization.
*/



SELECT MAX(pay_emp_salary) AS highest_salary, MIN(pay_emp_salary) AS lowest_salary FROM his_payrolls;
```



```sql
SELECT
    CASE
        WHEN pay_emp_salary < 30000 THEN 'Below 30K'
        WHEN pay_emp_salary BETWEEN 30000 AND 50000 THEN '30K-50K'
        WHEN pay_emp_salary BETWEEN 50000 AND 70000 THEN '50K-70K'
        ELSE 'Above 70K'
    END AS salary_range,
    COUNT(*) AS number_of_employees
FROM his_payrolls
GROUP BY salary_range;


/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
```



Payroll Aggregate Summary

Payroll Overview

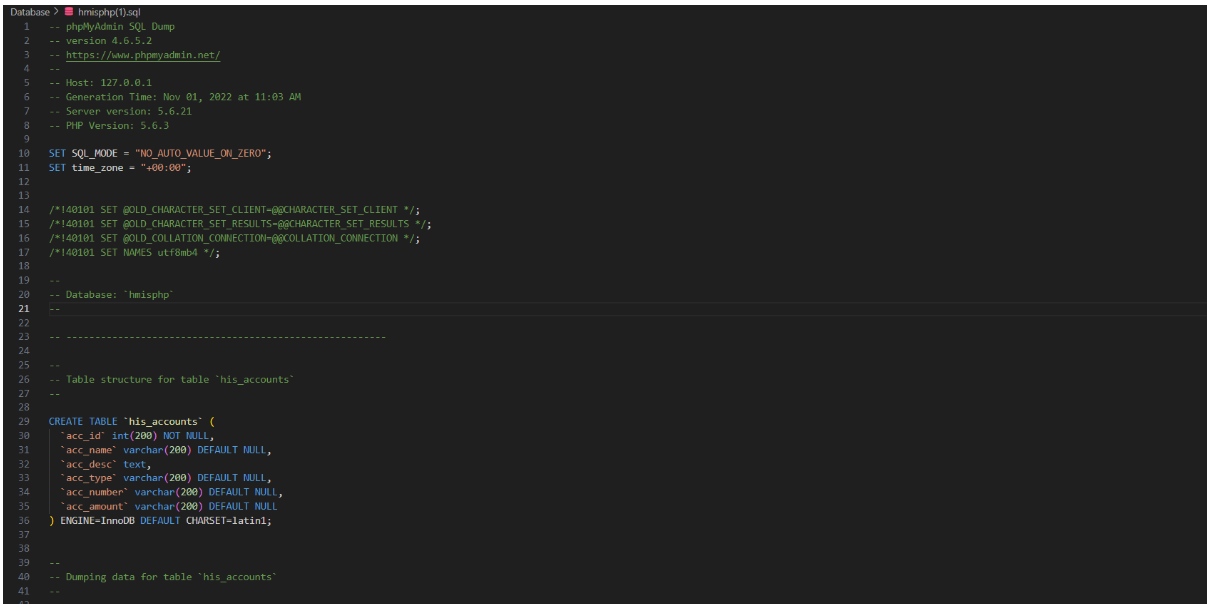
Total Payroll Amount	Average Salary	Total Employees	Highest Salary	Lowest Salary
Rs 27205	Rs 9068.3333333333	3	Rs 7555	Rs 15500

Salary Distribution

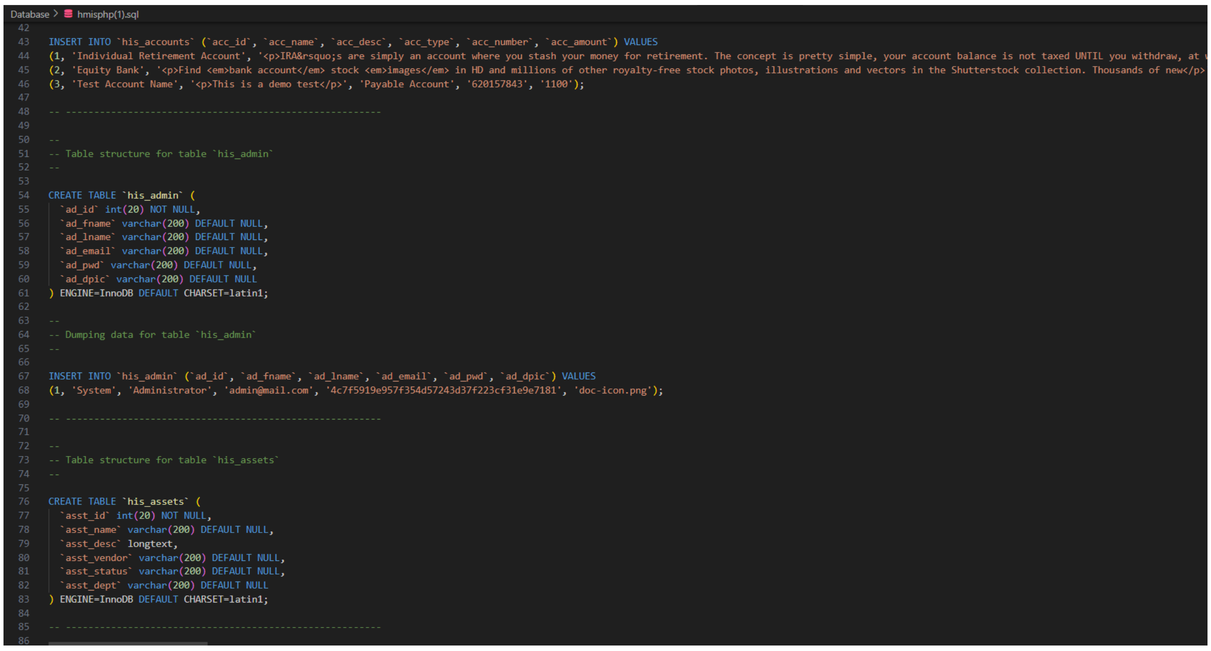
Salary Range	Number of Employees
Below 30K	3

2020 - 2024 © Hospital Management System.

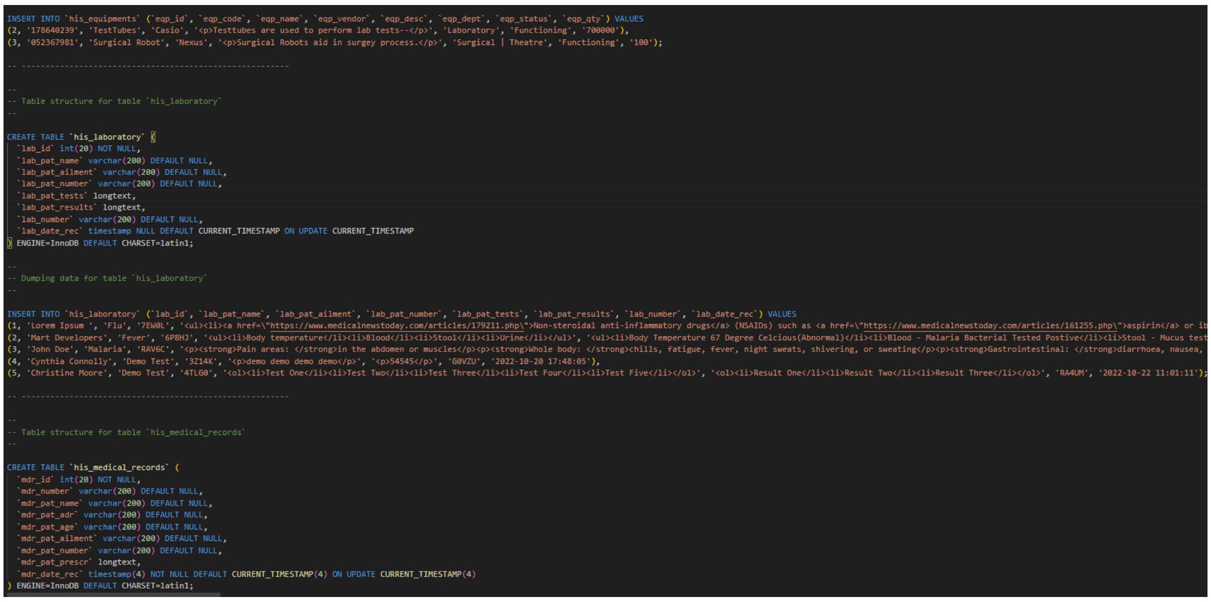
9. SQL queries (Create, Insert, Triggers, Procedures/ Functions, Nested query, Join, Aggregate queries) used in the project in the form of .sql file:



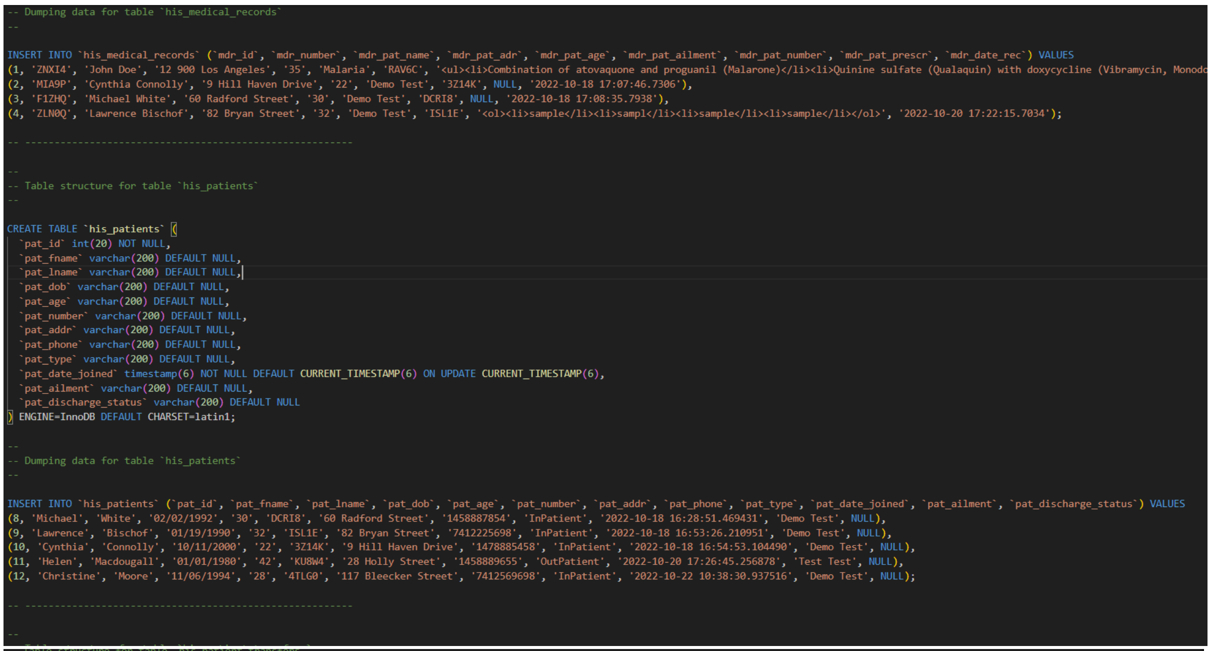
```sql
-- phpMyAdmin SQL Dump
-- version 4.6.5.2
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Nov 01, 2022 at 11:03 AM
-- Server version: 5.6.21
-- PHP Version: 5.6.3

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `hmisphp`
--

-- --------------------------------------------------------

--
-- Table structure for table `his_accounts`
--

CREATE TABLE `his_accounts` (
  `acc_id` int(200) NOT NULL,
  `acc_name` varchar(200) DEFAULT NULL,
  `acc_desc` text,
  `acc_type` varchar(200) DEFAULT NULL,
  `acc_number` varchar(200) DEFAULT NULL,
  `acc_amount` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


--
-- Dumping data for table `his_accounts`
--
```


Database > hmisphp(1).sql

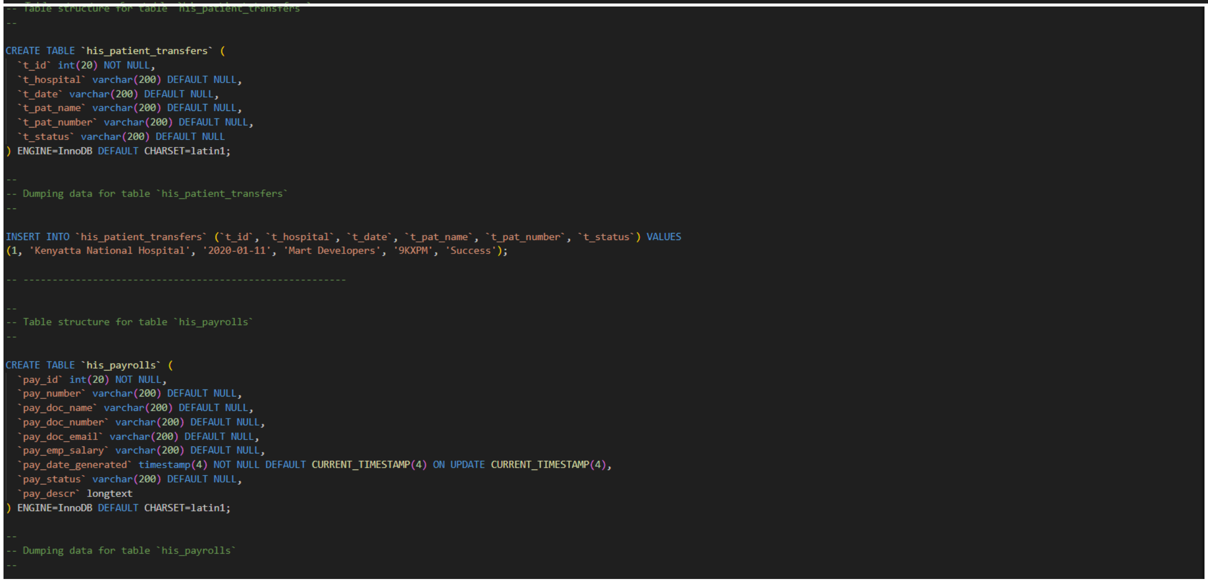
```sql

INSERT INTO `his_accounts` (`acc_id`, `acc_name`, `acc_desc`, `acc_type`, `acc_number`, `acc_amount`) VALUES
(1, 'Individual Retirement Account', '<p>IRA&rsquo;s are simply an account where you stash your money for retirement. The concept is pretty simple, your account balance is not taxed UNTIL you withdraw, at
(2, 'Equity Bank', '<p>Find <em>bank account</em> stock <em>images</em> in HD and millions of other royalty-free stock photos, illustrations and vectors in the Shutterstock collection. Thousands of new</p>
(3, 'Test Account Name', '<p>This is a demo test</p>', 'Payable Account', '620157843', '1100');

-- --------------------------------------------------------

--
-- Table structure for table `his_admin`
--

CREATE TABLE `his_admin` (
  `ad_id` int(20) NOT NULL,
  `ad_fname` varchar(200) DEFAULT NULL,
  `ad_lname` varchar(200) DEFAULT NULL,
  `ad_email` varchar(200) DEFAULT NULL,
  `ad_pwd` varchar(200) DEFAULT NULL,
  `ad_dpic` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `his_admin`
--

INSERT INTO `his_admin` (`ad_id`, `ad_fname`, `ad_lname`, `ad_email`, `ad_pwd`, `ad_dpic`) VALUES
(1, 'System', 'Administrator', 'admin@mail.com', '4c7f5919e957f354d57243d37f223cf31e9e7181', 'doc-icon.png');

-- --------------------------------------------------------

--
-- Table structure for table `his_assets`
--

CREATE TABLE `his_assets` (
  `asst_id` int(20) NOT NULL,
  `asst_name` varchar(200) DEFAULT NULL,
  `asst_desc` longtext,
  `asst_vendor` varchar(200) DEFAULT NULL,
  `asst_status` varchar(200) DEFAULT NULL,
  `asst_dept` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

```



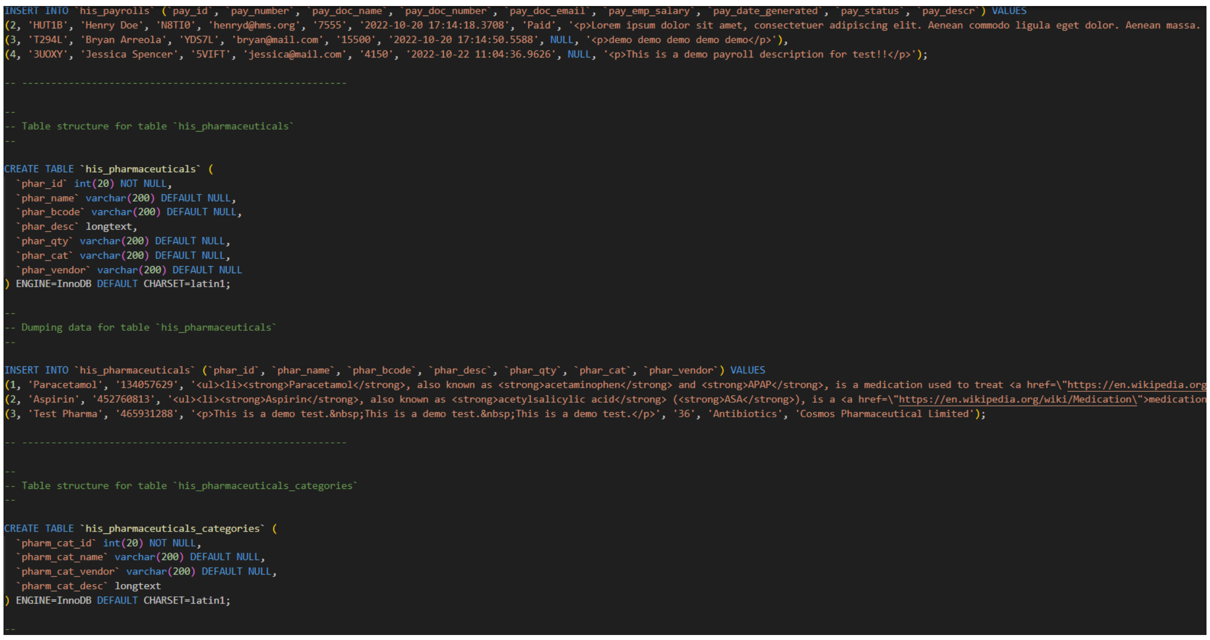
```sql
INSERT INTO `his_equipments` (`eqp_id`, `eqp_code`, `eqp_name`, `eqp_vendor`, `eqp_desc`, `eqp_dept`, `eqp_status`, `eqp_qty`) VALUES
(2, '178640239', 'TestTubes', 'Casio', '<p>Testtubes are used to perform lab tests--</p>', 'Laboratory', 'Functioning', '700000'),
(3, '052367981', 'Surgical Robot', 'Nexus', '<p>Surgical Robots aid in surgey process.</p>', 'Surgical | Theatre', 'Functioning', '100');

-- --------------------------------------------------------

--
-- Table structure for table `his_laboratory`
--

CREATE TABLE `his_laboratory` (
  `lab_id` int(20) NOT NULL,
  `lab_pat_name` varchar(200) DEFAULT NULL,
  `lab_pat_ailment` varchar(200) DEFAULT NULL,
  `lab_pat_number` varchar(200) DEFAULT NULL,
  `lab_pat_tests` longtext,
  `lab_pat_results` longtext,
  `lab_number` varchar(200) DEFAULT NULL,
  `lab_date_rec` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `his_laboratory`
--

INSERT INTO `his_laboratory` (`lab_id`, `lab_pat_name`, `lab_pat_ailment`, `lab_pat_number`, `lab_pat_tests`, `lab_pat_results`, `lab_number`, `lab_date_rec`) VALUES
(1, 'Lorem Ipsum ', 'Flu', '7EW0L', '<ul><li><a href=\"https://www.medicalnewstoday.com/articles/179211.php\">Non-steroidal anti-inflammatory drugs</a> (NSAIDs) such as <a href=\"https://www.medicalnewstoday.com/articles/161255.php\">aspirin</a> or ib
(2, 'Mart Developers', 'Fever', '6P8HJ', '<ul><li>Body temperature</li><li>Blood</li><li>Stool</li><li>Urine</li></ul>', '<ul><li>Body Temperature 67 Degree Celcious(Abnormal)</li><li>Blood - Malaria Bacterial Tested Postive</li><li>Stool - Mucus test
(3, 'John Doe', 'Malaria', 'RAV6C', '<p><strong>Pain areas: </strong>in the abdomen or muscles</p><p><strong>Whole body: </strong>chills, fatigue, fever, night sweats, shivering, or sweating</p><p><strong>Gastrointestinal: </strong>diarrhoea, nausea,
(4, 'Cynthia Connolly', 'Demo Test', '3Z14K', '<p>demo demo demo demo</p>', '<p>54545</p>', 'G0VZU', '2022-10-20 17:48:05'),
(5, 'Christine Moore', 'Demo Test', '4TLG0', '<ol><li>Test One</li><li>Test Two</li><li>Test Three</li><li>Test Four</li><li>Test Five</li></ol>', '<ol><li>Result One</li><li>Result Two</li><li>Result Three</li></ol>', 'RA4UM', '2022-10-22 11:01:11');

-- --------------------------------------------------------

--
-- Table structure for table `his_medical_records`
--

CREATE TABLE `his_medical_records` (
  `mdr_id` int(20) NOT NULL,
  `mdr_number` varchar(200) DEFAULT NULL,
  `mdr_pat_name` varchar(200) DEFAULT NULL,
  `mdr_pat_adr` varchar(200) DEFAULT NULL,
  `mdr_pat_age` varchar(200) DEFAULT NULL,
  `mdr_pat_ailment` varchar(200) DEFAULT NULL,
  `mdr_pat_number` varchar(200) DEFAULT NULL,
  `mdr_pat_prescr` longtext,
  `mdr_date_rec` timestamp(4) NOT NULL DEFAULT CURRENT_TIMESTAMP(4) ON UPDATE CURRENT_TIMESTAMP(4)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



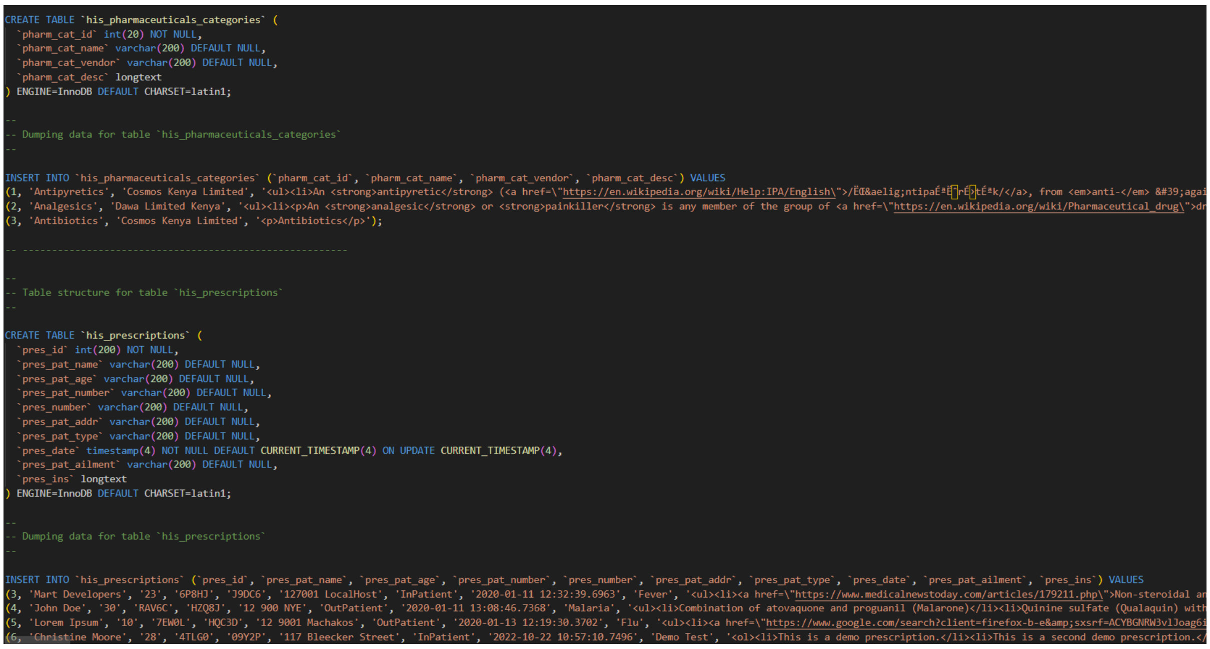
```
-- Dumping data for table `his_medical_records`
--

INSERT INTO `his_medical_records` (`mdr_id`, `mdr_number`, `mdr_pat_name`, `mdr_pat_adr`, `mdr_pat_age`, `mdr_pat_ailment`, `mdr_pat_number`, `mdr_pat_prescr`, `mdr_date_rec`) VALUES
(1, 'ZNXI4', 'John Doe', '12 900 Los Angeles', '35', 'Malaria', 'RAV6C', '<ul><li>Combination of atovaquone and proguanil (Malarone)</li><li>Quinine sulfate (Qualaquin) with doxycycline (Vibramycin, Monodo
(2, 'MIA9P', 'Cynthia Connolly', '9 Hill Haven Drive', '22', 'Demo Test', '3Z14K', NULL, '2022-10-18 17:07:46.7306'),
(3, 'F1ZHQ', 'Michael White', '60 Radford Street', '30', 'Demo Test', 'DCRI8', NULL, '2022-10-18 17:08:35.7938'),
(4, 'ZLN0Q', 'Lawrence Bischof', '82 Bryan Street', '32', 'Demo Test', 'ISL1E', '<ol><li>sample</li><li>sampl</li><li>sample</li><li>sample</li></ol>', '2022-10-20 17:22:15.7034');

-- --------------------------------------------------------

--
-- Table structure for table `his_patients`
--

CREATE TABLE `his_patients` (
  `pat_id` int(20) NOT NULL,
  `pat_fname` varchar(200) DEFAULT NULL,
  `pat_lname` varchar(200) DEFAULT NULL,
  `pat_dob` varchar(200) DEFAULT NULL,
  `pat_age` varchar(200) DEFAULT NULL,
  `pat_number` varchar(200) DEFAULT NULL,
  `pat_addr` varchar(200) DEFAULT NULL,
  `pat_phone` varchar(200) DEFAULT NULL,
  `pat_type` varchar(200) DEFAULT NULL,
  `pat_date_joined` timestamp(6) NOT NULL DEFAULT CURRENT_TIMESTAMP(6) ON UPDATE CURRENT_TIMESTAMP(6),
  `pat_ailment` varchar(200) DEFAULT NULL,
  `pat_discharge_status` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `his_patients`
--

INSERT INTO `his_patients` (`pat_id`, `pat_fname`, `pat_lname`, `pat_dob`, `pat_age`, `pat_number`, `pat_addr`, `pat_phone`, `pat_type`, `pat_date_joined`, `pat_ailment`, `pat_discharge_status`) VALUES
(8, 'Michael', 'White', '02/02/1992', '30', 'DCRI8', '60 Radford Street', '1458887854', 'InPatient', '2022-10-18 16:28:51.469431', 'Demo Test', NULL),
(9, 'Lawrence', 'Bischof', '01/19/1990', '32', 'ISL1E', '82 Bryan Street', '7412225698', 'InPatient', '2022-10-18 16:53:26.210951', 'Demo Test', NULL),
(10, 'Cynthia', 'Connolly', '10/11/2000', '22', '3Z14K', '9 Hill Haven Drive', '1478885458', 'InPatient', '2022-10-18 16:54:53.104490', 'Demo Test', NULL),
(11, 'Helen', 'Macdougall', '01/01/1980', '42', 'KU8W4', '28 Holly Street', '1458889655', 'OutPatient', '2022-10-20 17:26:45.256878', 'Test Test', NULL),
(12, 'Christine', 'Moore', '11/06/1994', '28', '4TLG0', '117 Bleecker Street', '7412569698', 'InPatient', '2022-10-22 10:38:30.937516', 'Demo Test', NULL);

-- --------------------------------------------------------

--
```



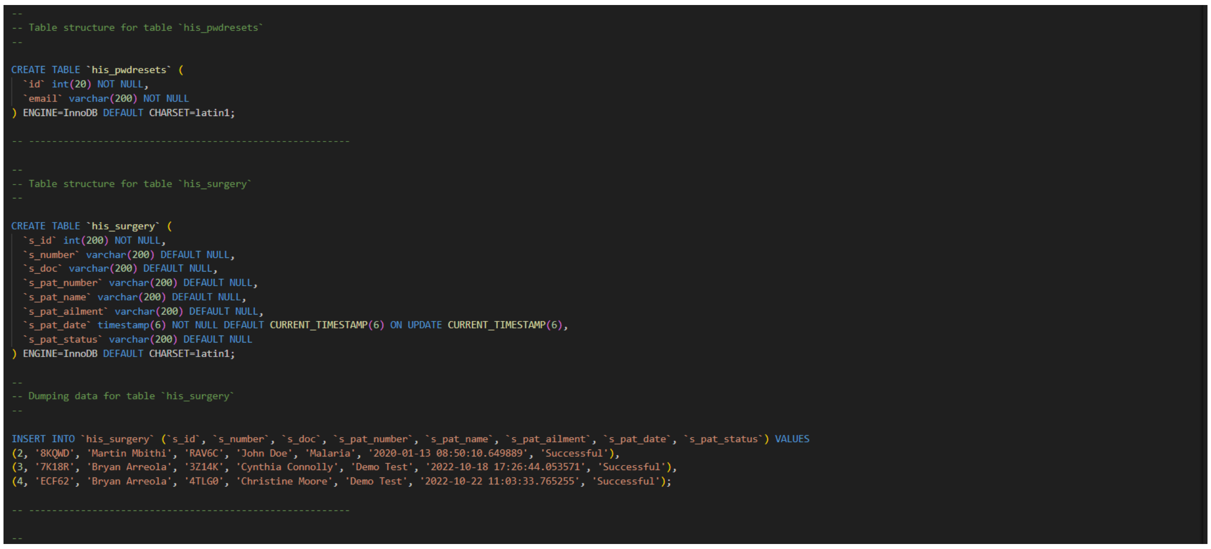
```
-- Table structure for table `his_patient_transfers`
--

CREATE TABLE `his_patient_transfers` (
  `t_id` int(20) NOT NULL,
  `t_hospital` varchar(200) DEFAULT NULL,
  `t_date` varchar(200) DEFAULT NULL,
  `t_pat_name` varchar(200) DEFAULT NULL,
  `t_pat_number` varchar(200) DEFAULT NULL,
  `t_status` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `his_patient_transfers`
--

INSERT INTO `his_patient_transfers` (`t_id`, `t_hospital`, `t_date`, `t_pat_name`, `t_pat_number`, `t_status`) VALUES
(1, 'Kenyatta National Hospital', '2020-01-11', 'Mart Developers', '9KXPM', 'Success');

-- --------------------------------------------------------

--
-- Table structure for table `his_payrolls`
--

CREATE TABLE `his_payrolls` (
  `pay_id` int(20) NOT NULL,
  `pay_number` varchar(200) DEFAULT NULL,
  `pay_doc_name` varchar(200) DEFAULT NULL,
  `pay_doc_number` varchar(200) DEFAULT NULL,
  `pay_doc_email` varchar(200) DEFAULT NULL,
  `pay_emp_salary` varchar(200) DEFAULT NULL,
  `pay_date_generated` timestamp(4) NOT NULL DEFAULT CURRENT_TIMESTAMP(4) ON UPDATE CURRENT_TIMESTAMP(4),
  `pay_status` varchar(200) DEFAULT NULL,
  `pay_descr` longtext
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `his_payrolls`
--
```



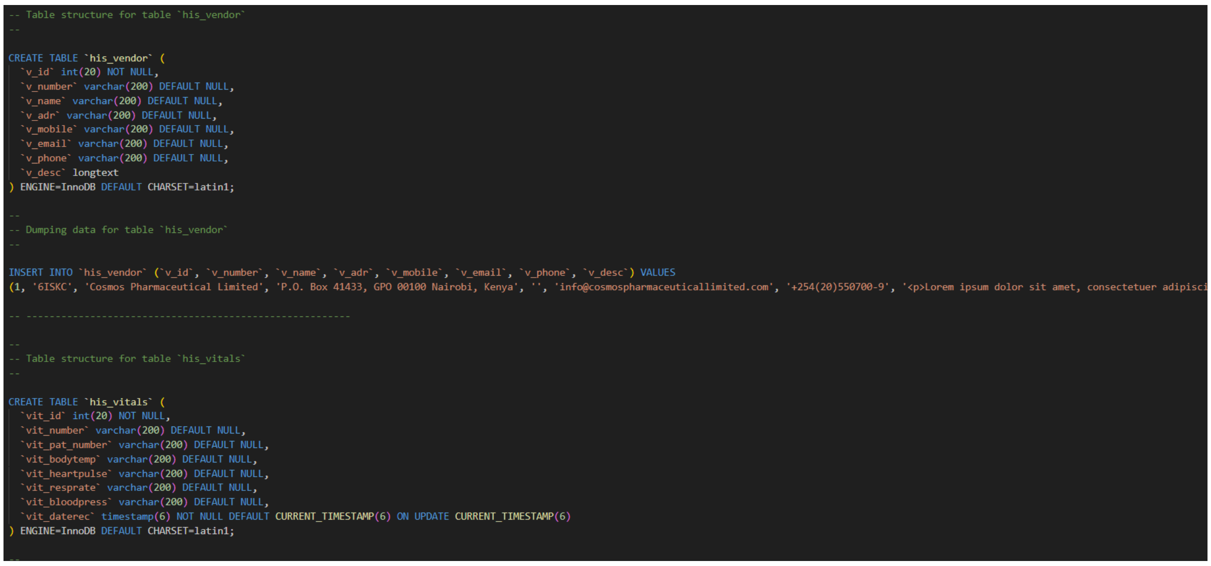
```sql
INSERT INTO `his_payrolls` (`pay_id`, `pay_number`, `pay_doc_name`, `pay_doc_number`, `pay_doc_email`, `pay_emp_salary`, `pay_date_generated`, `pay_status`, `pay_descr`) VALUES
(2, 'HUT1B', 'Henry Doe', 'N8TI0', 'henryd@hms.org', '7555', '2022-10-20 17:14:18.3708', 'Paid', '<p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa.
(3, 'T294L', 'Bryan Arreola', 'YDS7L', 'bryan@mail.com', '15500', '2022-10-20 17:14:50.5588', NULL, '<p>demo demo demo demo demo</p>'),
(4, '3UOXY', 'Jessica Spencer', '5VIFT', 'jessica@mail.com', '4150', '2022-10-22 11:04:36.9626', NULL, '<p>This is a demo payroll description for test!!</p>');

-- --------------------------------------------------------

--
-- Table structure for table `his_pharmaceuticals`
--

CREATE TABLE `his_pharmaceuticals` (
  `phar_id` int(20) NOT NULL,
  `phar_name` varchar(200) DEFAULT NULL,
  `phar_bcode` varchar(200) DEFAULT NULL,
  `phar_desc` longtext,
  `phar_qty` varchar(200) DEFAULT NULL,
  `phar_cat` varchar(200) DEFAULT NULL,
  `phar_vendor` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `his_pharmaceuticals`
--

INSERT INTO `his_pharmaceuticals` (`phar_id`, `phar_name`, `phar_bcode`, `phar_desc`, `phar_qty`, `phar_cat`, `phar_vendor`) VALUES
(1, 'Paracetamol', '134057629', '<ul><li><strong>Paracetamol</strong>, also known as <strong>acetaminophen</strong> and <strong>APAP</strong>, is a medication used to treat <a href=\"https://en.wikipedia.org
(2, 'Aspirin', '452760813', '<ul><li><strong>Aspirin</strong>, also known as <strong>acetylsalicylic acid</strong> (<strong>ASA</strong>), is a <a href=\"https://en.wikipedia.org/wiki/Medication\">medication
(3, 'Test Pharma', '465931288', '<p>This is a demo test.&nbsp;This is a demo test.&nbsp;This is a demo test.</p>', '36', 'Antibiotics', 'Cosmos Pharmaceutical Limited');

-- --------------------------------------------------------

--
-- Table structure for table `his_pharmaceuticals_categories`
--

CREATE TABLE `his_pharmaceuticals_categories` (
  `pharm_cat_id` int(20) NOT NULL,
  `pharm_cat_name` varchar(200) DEFAULT NULL,
  `pharm_cat_vendor` varchar(200) DEFAULT NULL,
  `pharm_cat_desc` longtext
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
```



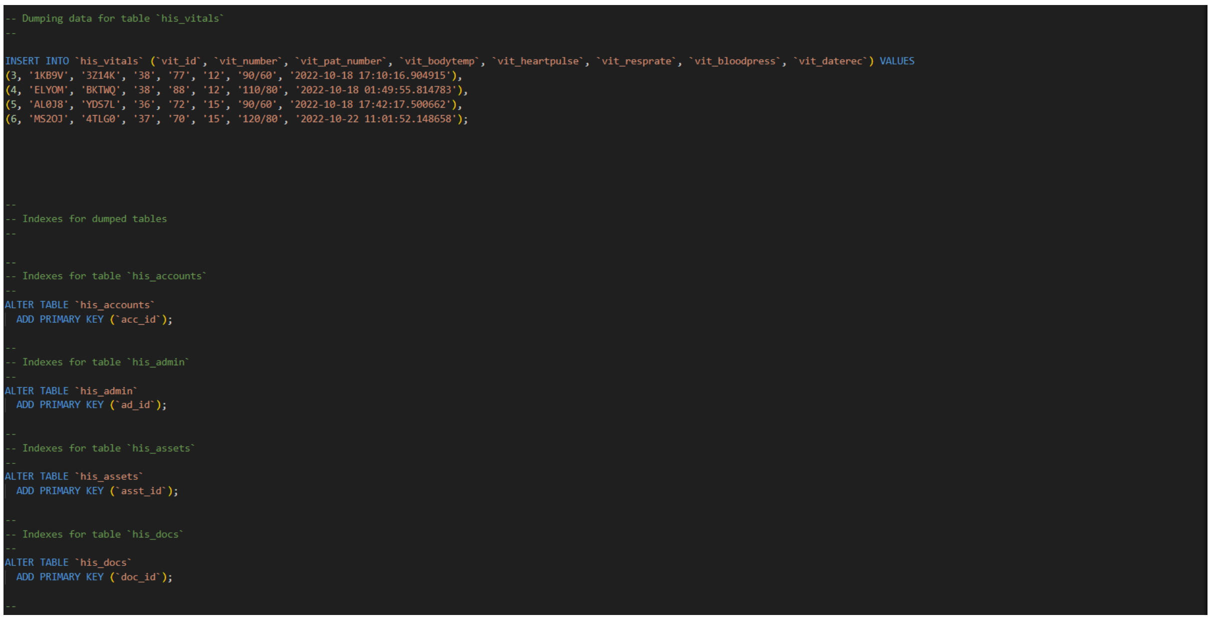
```sql
CREATE TABLE `his_pharmaceuticals_categories` (
  `pharm_cat_id` int(20) NOT NULL,
  `pharm_cat_name` varchar(200) DEFAULT NULL,
  `pharm_cat_vendor` varchar(200) DEFAULT NULL,
  `pharm_cat_desc` longtext
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `his_pharmaceuticals_categories`
--

INSERT INTO `his_pharmaceuticals_categories` (`pharm_cat_id`, `pharm_cat_name`, `pharm_cat_vendor`, `pharm_cat_desc`) VALUES
(1, 'Antipyretics', 'Cosmos Kenya Limited', '<ul><li>An <strong>antipyretic</strong> (<a href=\"https://en.wikipedia.org/wiki/Help:IPA/English\">/ËŒ&aelig;ntipaÉªËˆrÉ›tÉªk/</a>, from <em>anti-</em> &#39;agai
(2, 'Analgesics', 'Dawa Limited Kenya', '<ul><li><p>An <strong>analgesic</strong> or <strong>painkiller</strong> is any member of the group of <a href=\"https://en.wikipedia.org/wiki/Pharmaceutical_drug\">dr
(3, 'Antibiotics', 'Cosmos Kenya Limited', '<p>Antibiotics</p>');

-- --------------------------------------------------------

--
-- Table structure for table `his_prescriptions`
--

CREATE TABLE `his_prescriptions` (
  `pres_id` int(200) NOT NULL,
  `pres_pat_name` varchar(200) DEFAULT NULL,
  `pres_pat_age` varchar(200) DEFAULT NULL,
  `pres_pat_number` varchar(200) DEFAULT NULL,
  `pres_number` varchar(200) DEFAULT NULL,
  `pres_pat_addr` varchar(200) DEFAULT NULL,
  `pres_pat_type` varchar(200) DEFAULT NULL,
  `pres_date` timestamp(4) NOT NULL DEFAULT CURRENT_TIMESTAMP(4) ON UPDATE CURRENT_TIMESTAMP(4),
  `pres_pat_ailment` varchar(200) DEFAULT NULL,
  `pres_ins` longtext
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `his_prescriptions`
--

INSERT INTO `his_prescriptions` (`pres_id`, `pres_pat_name`, `pres_pat_age`, `pres_pat_number`, `pres_number`, `pres_pat_addr`, `pres_pat_type`, `pres_date`, `pres_pat_ailment`, `pres_ins`) VALUES
(3, 'Mart Developers', '23', '6P8HJ', 'J9DC6', '127001 LocalHost', 'InPatient', '2020-01-11 12:32:39.6963', 'Fever', '<ul><li><a href=\"https://www.medicalnewstoday.com/articles/179211.php\">Non-steroidal an
(4, 'John Doe', '30', 'RAV6C', 'HZQ8J', '12 900 NYE', 'OutPatient', '2020-01-11 13:08:46.7368', 'Malaria', '<ul><li>Combination of atovaquone and proguanil (Malarone)</li><li>Quinine sulfate (Qualaquin) with
(5, 'Lorem Ipsum', '10', '7EW0L', 'HQC3D', '12 9001 Machakos', 'OutPatient', '2020-01-13 12:19:30.3702', 'Flu', '<ul><li><a href=\"https://www.google.com/search?client=firefox-b-e&amp;sxsrf=ACYBGNRW3vlJoag6i
(6, 'Christine Moore', '28', '4TLG0', '09Y2P', '117 Bleecker Street', 'InPatient', '2022-10-22 10:57:10.7496', 'Demo Test', '<ol><li>This is a demo prescription.</li><li>This is a second demo prescription.</
```



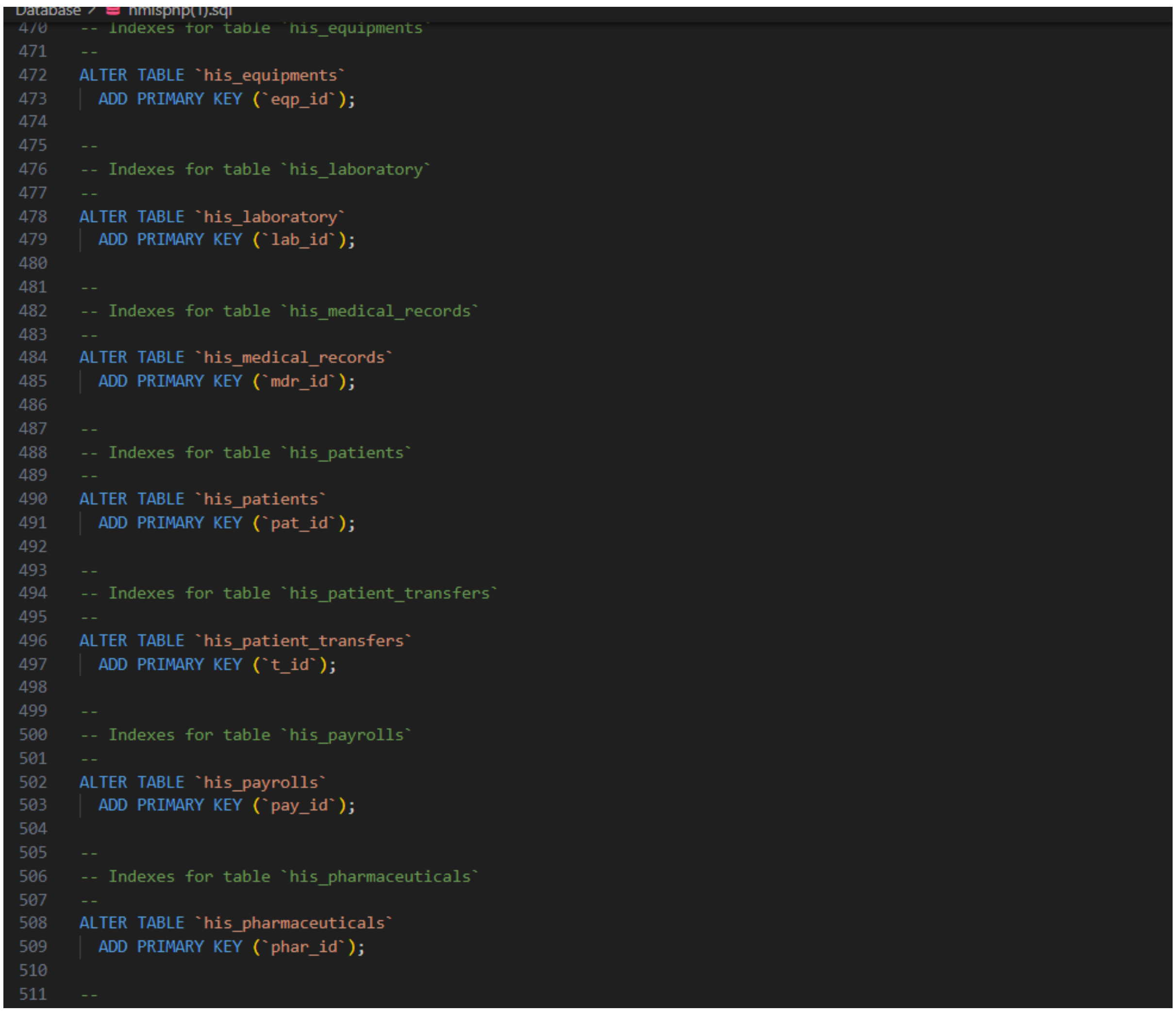
```sql
--
-- Table structure for table `his_pwdresets`
--

CREATE TABLE `his_pwdresets` (
  `id` int(20) NOT NULL,
  `email` varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `his_surgery`
--

CREATE TABLE `his_surgery` (
  `s_id` int(200) NOT NULL,
  `s_number` varchar(200) DEFAULT NULL,
  `s_doc` varchar(200) DEFAULT NULL,
  `s_pat_number` varchar(200) DEFAULT NULL,
  `s_pat_name` varchar(200) DEFAULT NULL,
  `s_pat_ailment` varchar(200) DEFAULT NULL,
  `s_pat_date` timestamp(6) NOT NULL DEFAULT CURRENT_TIMESTAMP(6) ON UPDATE CURRENT_TIMESTAMP(6),
  `s_pat_status` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `his_surgery`
--

INSERT INTO `his_surgery` (`s_id`, `s_number`, `s_doc`, `s_pat_number`, `s_pat_name`, `s_pat_ailment`, `s_pat_date`, `s_pat_status`) VALUES
(2, '8KQWD', 'Martin Mbithi', 'RAV6C', 'John Doe', 'Malaria', '2020-01-13 08:50:10.649889', 'Successful'),
(3, '7K18R', 'Bryan Arreola', '3Z14K', 'Cynthia Connolly', 'Demo Test', '2022-10-18 17:26:44.053571', 'Successful'),
(4, 'ECF62', 'Bryan Arreola', '4TLG0', 'Christine Moore', 'Demo Test', '2022-10-22 11:03:33.765255', 'Successful');

-- --------------------------------------------------------

--
```



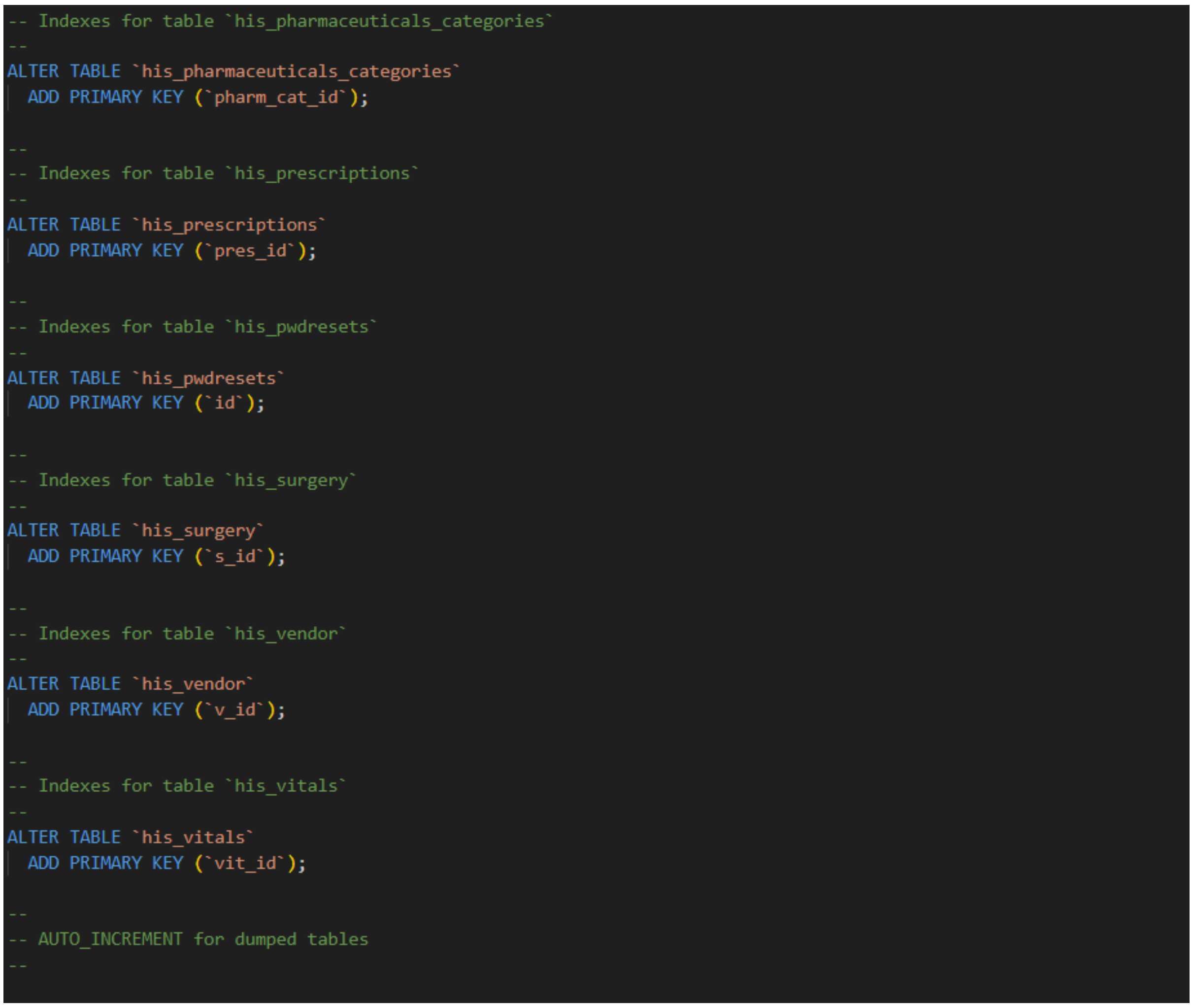
```sql
-- Table structure for table `his_vendor`
--

CREATE TABLE `his_vendor` (
  `v_id` int(20) NOT NULL,
  `v_number` varchar(200) DEFAULT NULL,
  `v_name` varchar(200) DEFAULT NULL,
  `v_adr` varchar(200) DEFAULT NULL,
  `v_mobile` varchar(200) DEFAULT NULL,
  `v_email` varchar(200) DEFAULT NULL,
  `v_phone` varchar(200) DEFAULT NULL,
  `v_desc` longtext
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `his_vendor`
--

INSERT INTO `his_vendor` (`v_id`, `v_number`, `v_name`, `v_adr`, `v_mobile`, `v_email`, `v_phone`, `v_desc`) VALUES
(1, '6ISKC', 'Cosmos Pharmaceutical Limited', 'P.O. Box 41433, GPO 00100 Nairobi, Kenya', '', 'info@cosmospharmaceuticallimited.com', '+254(20)550700-9', '<p>Lorem ipsum dolor sit amet, consectetuer adipisci

-- --------------------------------------------------------

--
-- Table structure for table `his_vitals`
--

CREATE TABLE `his_vitals` (
  `vit_id` int(20) NOT NULL,
  `vit_number` varchar(200) DEFAULT NULL,
  `vit_pat_number` varchar(200) DEFAULT NULL,
  `vit_bodytemp` varchar(200) DEFAULT NULL,
  `vit_heartpulse` varchar(200) DEFAULT NULL,
  `vit_resprate` varchar(200) DEFAULT NULL,
  `vit_bloodpress` varchar(200) DEFAULT NULL,
  `vit_daterec` timestamp(6) NOT NULL DEFAULT CURRENT_TIMESTAMP(6) ON UPDATE CURRENT_TIMESTAMP(6)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
```

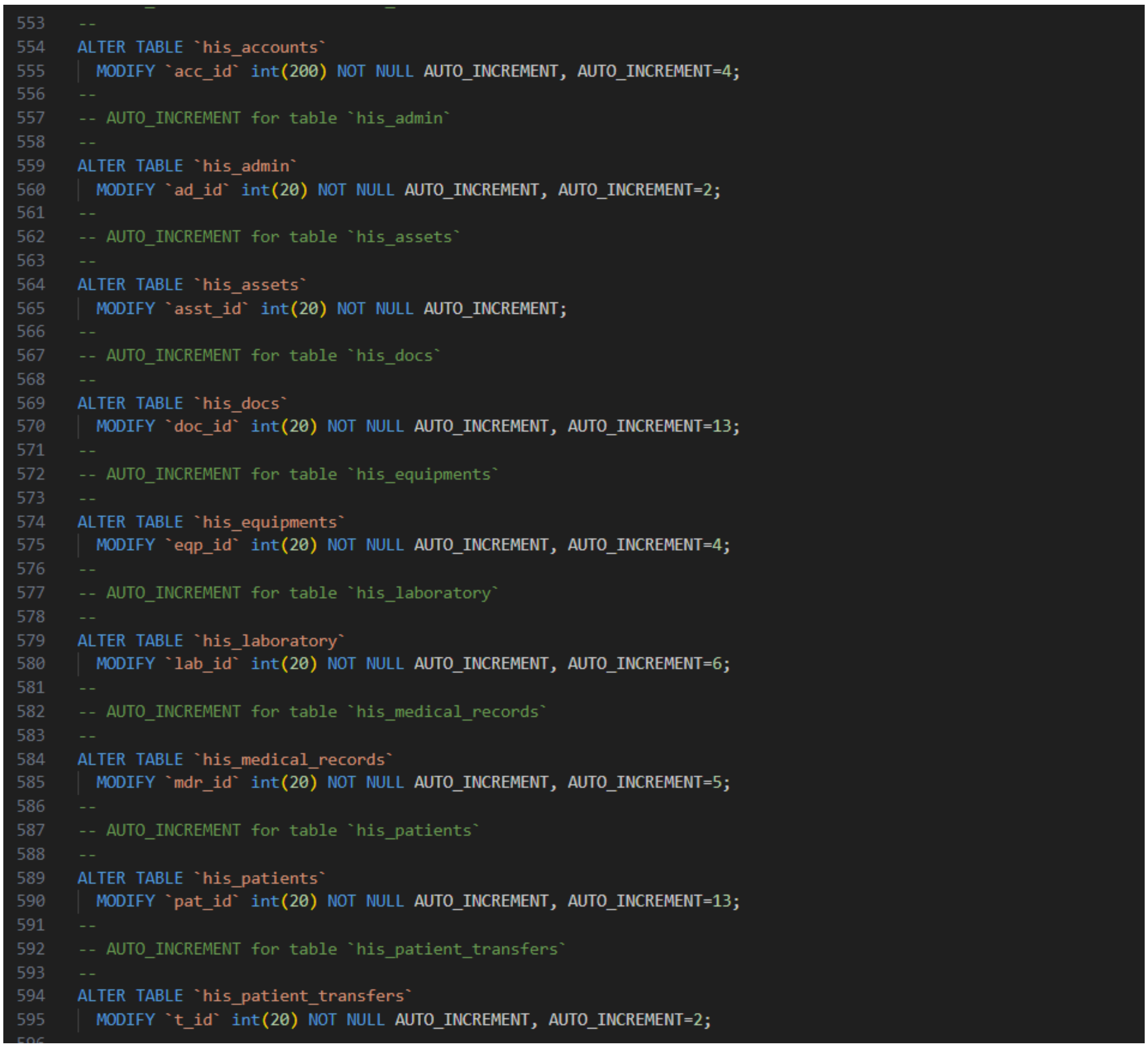


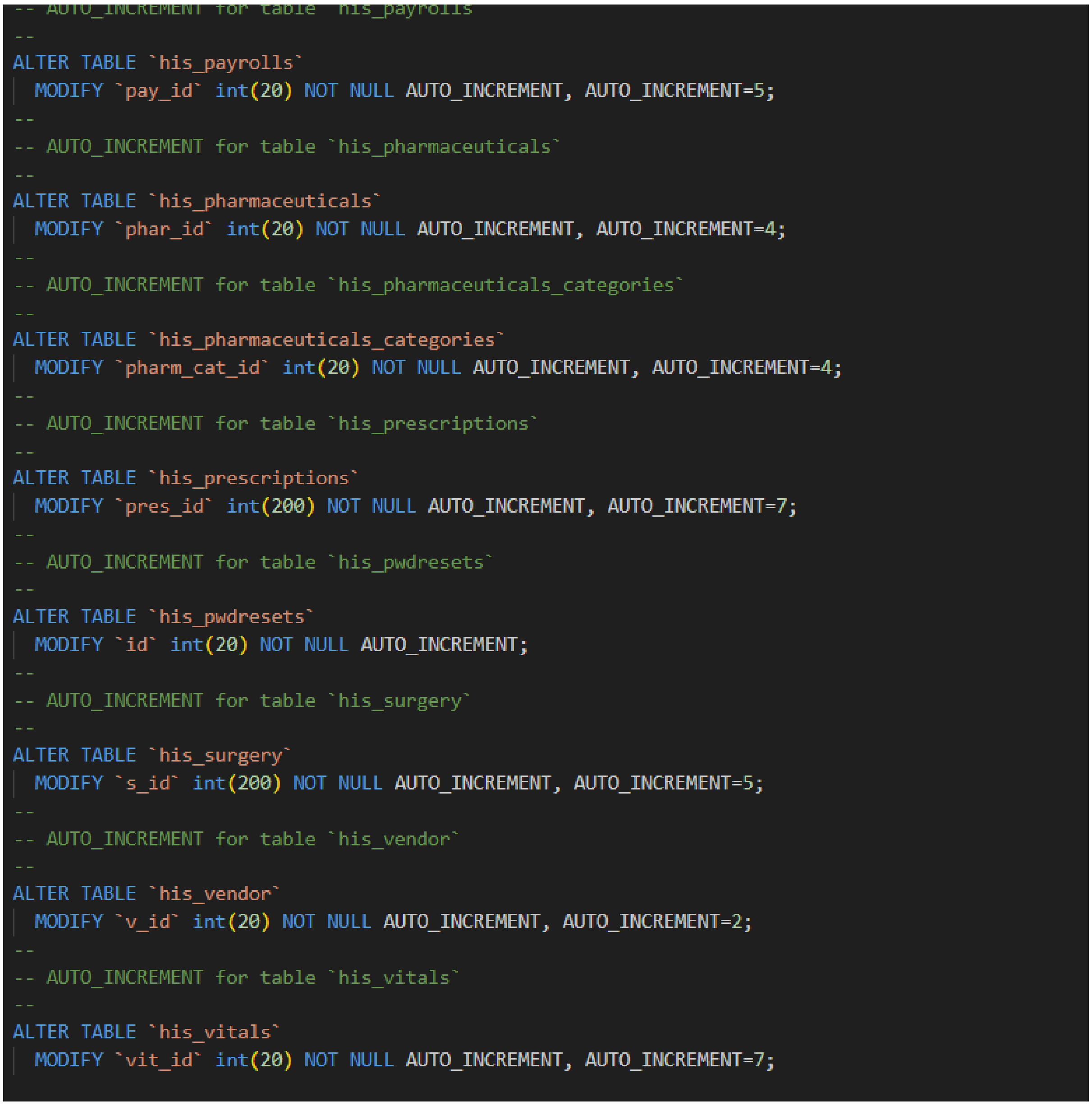
```sql
-- Dumping data for table `his_vitals`
--

INSERT INTO `his_vitals` (`vit_id`, `vit_number`, `vit_pat_number`, `vit_bodytemp`, `vit_heartpulse`, `vit_resprate`, `vit_bloodpress`, `vit_daterec`) VALUES
(3, '1KB9V', '3Z14K', '38', '77', '12', '90/60', '2022-10-18 17:10:16.904915'),
(4, 'ELYOM', 'BKTWQ', '38', '88', '12', '110/80', '2022-10-18 01:49:55.814783'),
(5, 'AL0J8', 'YDS7L', '36', '72', '15', '90/60', '2022-10-18 17:42:17.500662'),
(6, 'MS2OJ', '4TLG0', '37', '70', '15', '120/80', '2022-10-22 11:01:52.148658');





--
-- Indexes for dumped tables
--

--
-- Indexes for table `his_accounts`
--
ALTER TABLE `his_accounts`
  ADD PRIMARY KEY (`acc_id`);

--
-- Indexes for table `his_admin`
--
ALTER TABLE `his_admin`
  ADD PRIMARY KEY (`ad_id`);

--
-- Indexes for table `his_assets`
--
ALTER TABLE `his_assets`
  ADD PRIMARY KEY (`asst_id`);

--
-- Indexes for table `his_docs`
--
ALTER TABLE `his_docs`
  ADD PRIMARY KEY (`doc_id`);

--
```


Database > hmisphp(1).sql

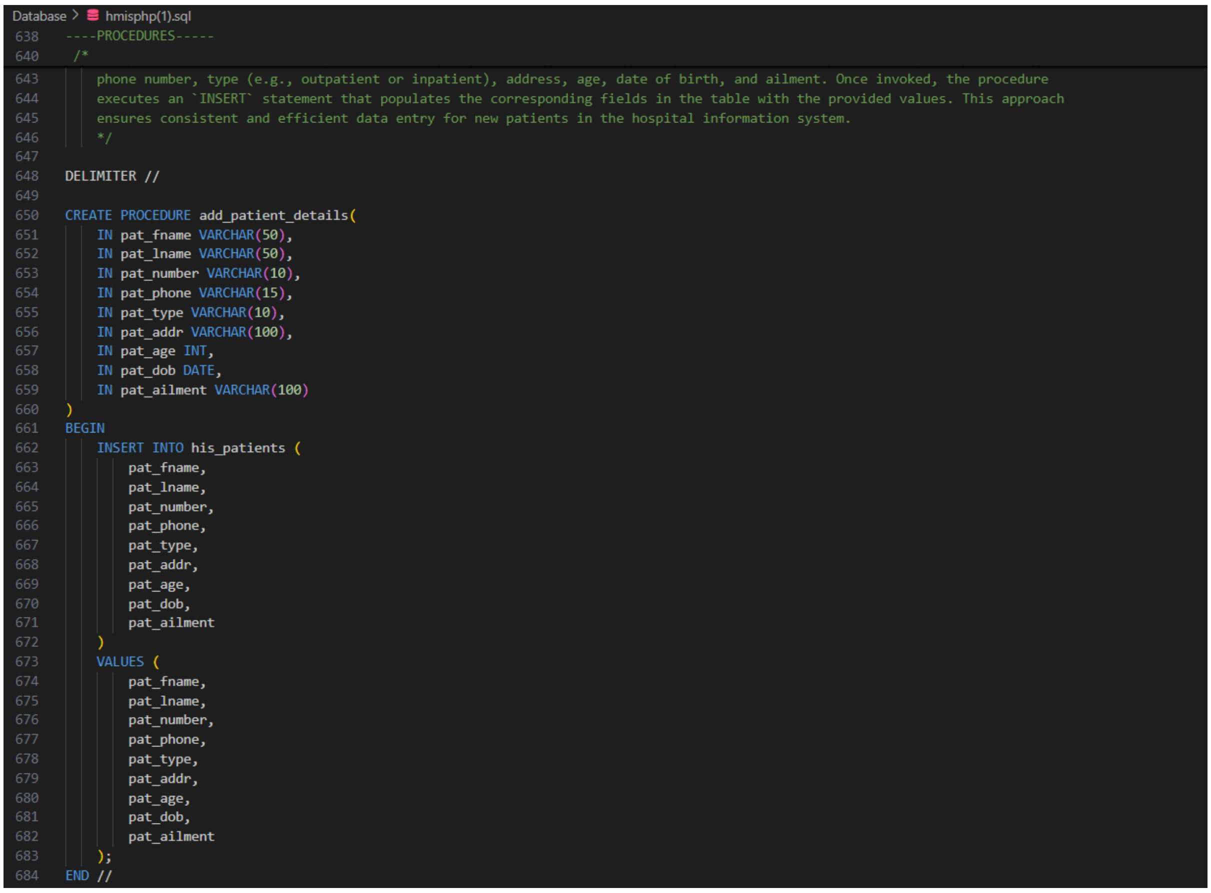
```sql
-- Indexes for table `his_equipments`
--
ALTER TABLE `his_equipments`
  ADD PRIMARY KEY (`eqp_id`);

--
-- Indexes for table `his_laboratory`
--
ALTER TABLE `his_laboratory`
  ADD PRIMARY KEY (`lab_id`);

--
-- Indexes for table `his_medical_records`
--
ALTER TABLE `his_medical_records`
  ADD PRIMARY KEY (`mdr_id`);

--
-- Indexes for table `his_patients`
--
ALTER TABLE `his_patients`
  ADD PRIMARY KEY (`pat_id`);

--
-- Indexes for table `his_patient_transfers`
--
ALTER TABLE `his_patient_transfers`
  ADD PRIMARY KEY (`t_id`);

--
-- Indexes for table `his_payrolls`
--
ALTER TABLE `his_payrolls`
  ADD PRIMARY KEY (`pay_id`);

--
-- Indexes for table `his_pharmaceuticals`
--
ALTER TABLE `his_pharmaceuticals`
  ADD PRIMARY KEY (`phar_id`);

--
```


```
-- Indexes for table `his_pharmaceuticals_categories`
--
ALTER TABLE `his_pharmaceuticals_categories`
  ADD PRIMARY KEY (`pharm_cat_id`);

--
-- Indexes for table `his_prescriptions`
--
ALTER TABLE `his_prescriptions`
  ADD PRIMARY KEY (`pres_id`);

--
-- Indexes for table `his_pwdresets`
--
ALTER TABLE `his_pwdresets`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `his_surgery`
--
ALTER TABLE `his_surgery`
  ADD PRIMARY KEY (`s_id`);

--
-- Indexes for table `his_vendor`
--
ALTER TABLE `his_vendor`
  ADD PRIMARY KEY (`v_id`);

--
-- Indexes for table `his_vitals`
--
ALTER TABLE `his_vitals`
  ADD PRIMARY KEY (`vit_id`);

--
-- AUTO_INCREMENT for dumped tables
--
```


```sql
--
ALTER TABLE `his_accounts`
  MODIFY `acc_id` int(200) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;
--
-- AUTO_INCREMENT for table `his_admin`
--
ALTER TABLE `his_admin`
  MODIFY `ad_id` int(20) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
--
-- AUTO_INCREMENT for table `his_assets`
--
ALTER TABLE `his_assets`
  MODIFY `asst_id` int(20) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `his_docs`
--
ALTER TABLE `his_docs`
  MODIFY `doc_id` int(20) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=13;
--
-- AUTO_INCREMENT for table `his_equipments`
--
ALTER TABLE `his_equipments`
  MODIFY `eqp_id` int(20) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;
--
-- AUTO_INCREMENT for table `his_laboratory`
--
ALTER TABLE `his_laboratory`
  MODIFY `lab_id` int(20) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;
--
-- AUTO_INCREMENT for table `his_medical_records`
--
ALTER TABLE `his_medical_records`
  MODIFY `mdr_id` int(20) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;
--
-- AUTO_INCREMENT for table `his_patients`
--
ALTER TABLE `his_patients`
  MODIFY `pat_id` int(20) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=13;
--
-- AUTO_INCREMENT for table `his_patient_transfers`
--
ALTER TABLE `his_patient_transfers`
  MODIFY `t_id` int(20) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
```

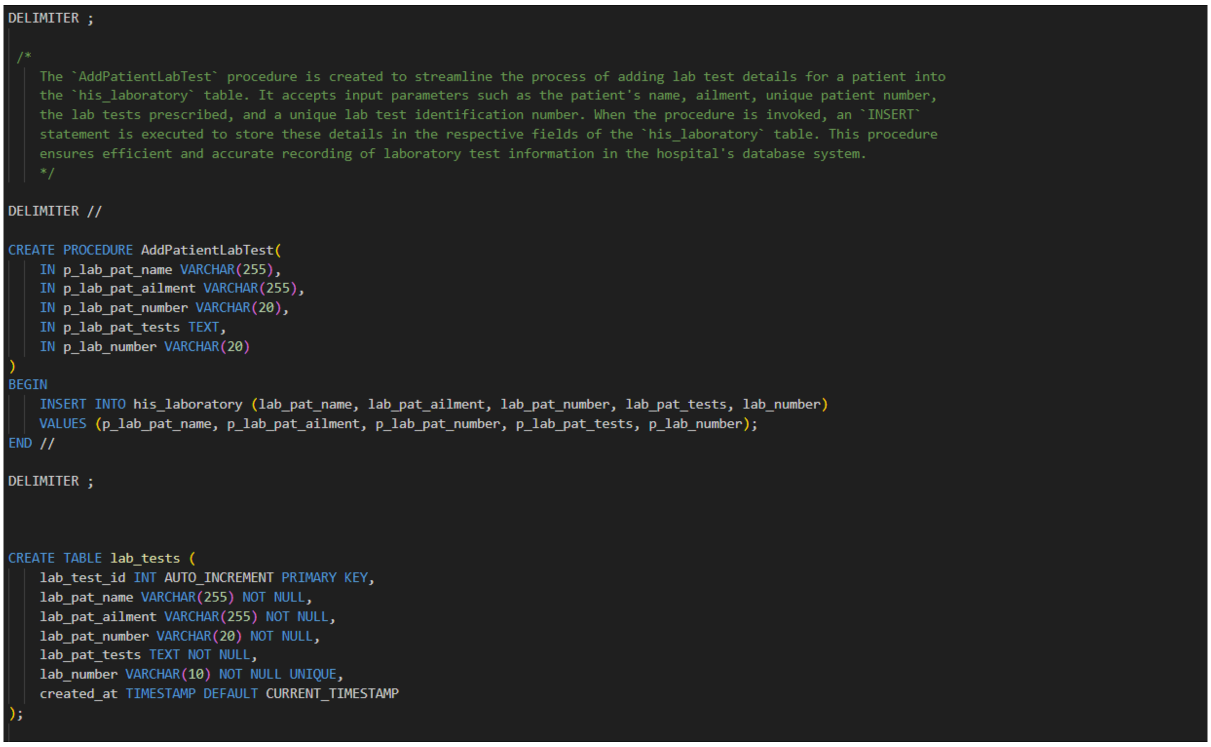


```sql
-- AUTO_INCREMENT for table `his_payrolls`
--
ALTER TABLE `his_payrolls`
  MODIFY `pay_id` int(20) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;
--
-- AUTO_INCREMENT for table `his_pharmaceuticals`
--
ALTER TABLE `his_pharmaceuticals`
  MODIFY `phar_id` int(20) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;
--
-- AUTO_INCREMENT for table `his_pharmaceuticals_categories`
--
ALTER TABLE `his_pharmaceuticals_categories`
  MODIFY `pharm_cat_id` int(20) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;
--
-- AUTO_INCREMENT for table `his_prescriptions`
--
ALTER TABLE `his_prescriptions`
  MODIFY `pres_id` int(200) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;
--
-- AUTO_INCREMENT for table `his_pwdresets`
--
ALTER TABLE `his_pwdresets`
  MODIFY `id` int(20) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `his_surgery`
--
ALTER TABLE `his_surgery`
  MODIFY `s_id` int(200) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;
--
-- AUTO_INCREMENT for table `his_vendor`
--
ALTER TABLE `his_vendor`
  MODIFY `v_id` int(20) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
--
-- AUTO_INCREMENT for table `his_vitals`
--
ALTER TABLE `his_vitals`
  MODIFY `vit_id` int(20) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;
```

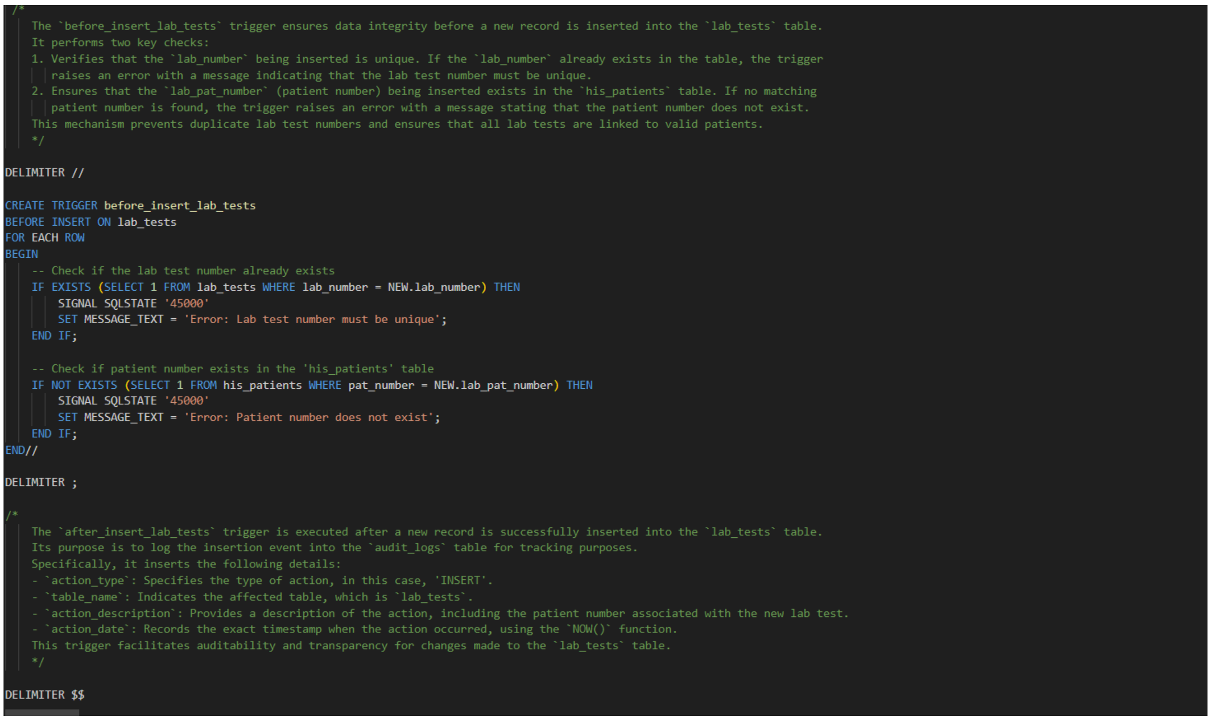


Database > hmisphp(1).sql

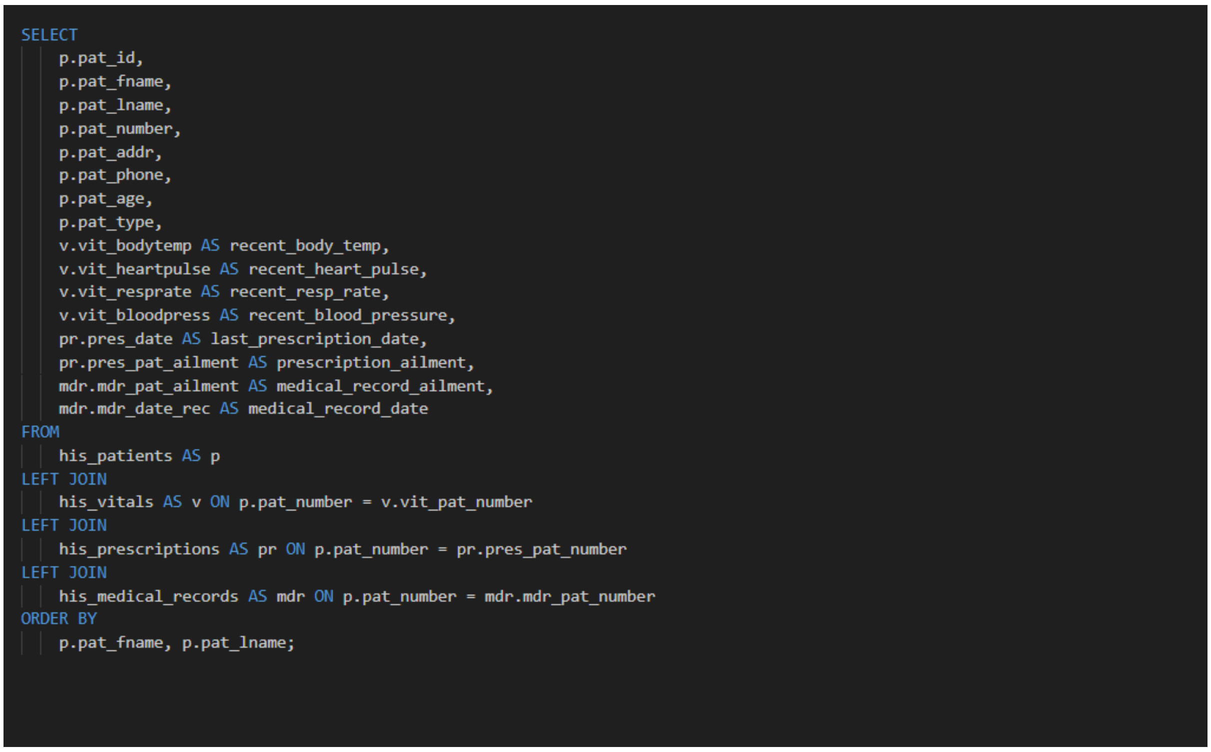
```
----PROCEDURES-----
 /*
    phone number, type (e.g., outpatient or inpatient), address, age, date of birth, and ailment. Once invoked, the procedure
    executes an `INSERT` statement that populates the corresponding fields in the table with the provided values. This approach
    ensures consistent and efficient data entry for new patients in the hospital information system.
    */

DELIMITER //

CREATE PROCEDURE add_patient_details(
    IN pat_fname VARCHAR(50),
    IN pat_lname VARCHAR(50),
    IN pat_number VARCHAR(10),
    IN pat_phone VARCHAR(15),
    IN pat_type VARCHAR(10),
    IN pat_addr VARCHAR(100),
    IN pat_age INT,
    IN pat_dob DATE,
    IN pat_ailment VARCHAR(100)
)
BEGIN
    INSERT INTO his_patients (
        pat_fname,
        pat_lname,
        pat_number,
        pat_phone,
        pat_type,
        pat_addr,
        pat_age,
        pat_dob,
        pat_ailment
    )
    VALUES (
        pat_fname,
        pat_lname,
        pat_number,
        pat_phone,
        pat_type,
        pat_addr,
        pat_age,
        pat_dob,
        pat_ailment
    );
END //
```



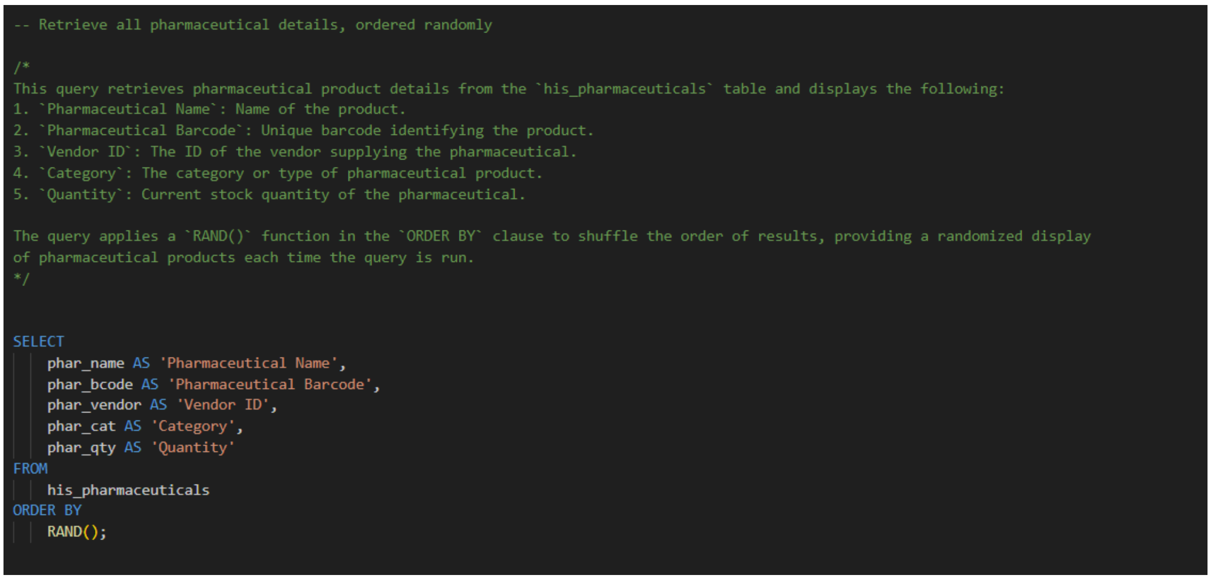
```
DELIMITER ;

 /*
    The `AddPatientLabTest` procedure is created to streamline the process of adding lab test details for a patient into
    the `his_laboratory` table. It accepts input parameters such as the patient's name, ailment, unique patient number,
    the lab tests prescribed, and a unique lab test identification number. When the procedure is invoked, an `INSERT`
    statement is executed to store these details in the respective fields of the `his_laboratory` table. This procedure
    ensures efficient and accurate recording of laboratory test information in the hospital's database system.
    */

DELIMITER //

CREATE PROCEDURE AddPatientLabTest(
    IN p_lab_pat_name VARCHAR(255),
    IN p_lab_pat_ailment VARCHAR(255),
    IN p_lab_pat_number VARCHAR(20),
    IN p_lab_pat_tests TEXT,
    IN p_lab_number VARCHAR(20)
)
BEGIN
    INSERT INTO his_laboratory (lab_pat_name, lab_pat_ailment, lab_pat_number, lab_pat_tests, lab_number)
    VALUES (p_lab_pat_name, p_lab_pat_ailment, p_lab_pat_number, p_lab_pat_tests, p_lab_number);
END //

DELIMITER ;



CREATE TABLE lab_tests (
    lab_test_id INT AUTO_INCREMENT PRIMARY KEY,
    lab_pat_name VARCHAR(255) NOT NULL,
    lab_pat_ailment VARCHAR(255) NOT NULL,
    lab_pat_number VARCHAR(20) NOT NULL,
    lab_pat_tests TEXT NOT NULL,
    lab_number VARCHAR(10) NOT NULL UNIQUE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```



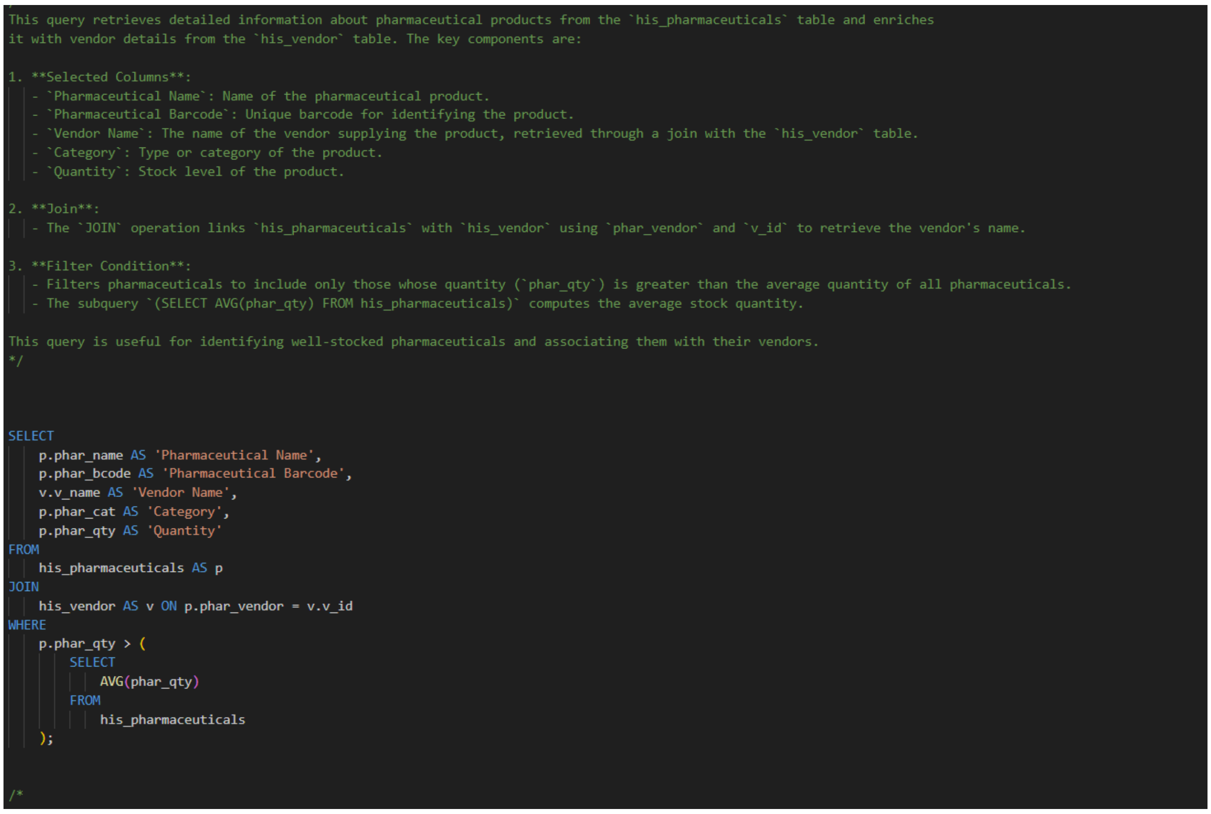
```sql
/*
  The `before_insert_lab_tests` trigger ensures data integrity before a new record is inserted into the `lab_tests` table.
  It performs two key checks:
  1. Verifies that the `lab_number` being inserted is unique. If the `lab_number` already exists in the table, the trigger
     raises an error with a message indicating that the lab test number must be unique.
  2. Ensures that the `lab_pat_number` (patient number) being inserted exists in the `his_patients` table. If no matching
     patient number is found, the trigger raises an error with a message stating that the patient number does not exist.
  This mechanism prevents duplicate lab test numbers and ensures that all lab tests are linked to valid patients.
  */

DELIMITER //

CREATE TRIGGER before_insert_lab_tests
BEFORE INSERT ON lab_tests
FOR EACH ROW
BEGIN
    -- Check if the lab test number already exists
    IF EXISTS (SELECT 1 FROM lab_tests WHERE lab_number = NEW.lab_number) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Error: Lab test number must be unique';
    END IF;

    -- Check if patient number exists in the 'his_patients' table
    IF NOT EXISTS (SELECT 1 FROM his_patients WHERE pat_number = NEW.lab_pat_number) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Error: Patient number does not exist';
    END IF;
END//

DELIMITER ;

/*
    The `after_insert_lab_tests` trigger is executed after a new record is successfully inserted into the `lab_tests` table.
    Its purpose is to log the insertion event into the `audit_logs` table for tracking purposes.
    Specifically, it inserts the following details:
    - `action_type`: Specifies the type of action, in this case, 'INSERT'.
    - `table_name`: Indicates the affected table, which is `lab_tests`.
    - `action_description`: Provides a description of the action, including the patient number associated with the new lab test.
    - `action_date`: Records the exact timestamp when the action occurred, using the `NOW()` function.
    This trigger facilitates auditability and transparency for changes made to the `lab_tests` table.
    */

DELIMITER $$
```



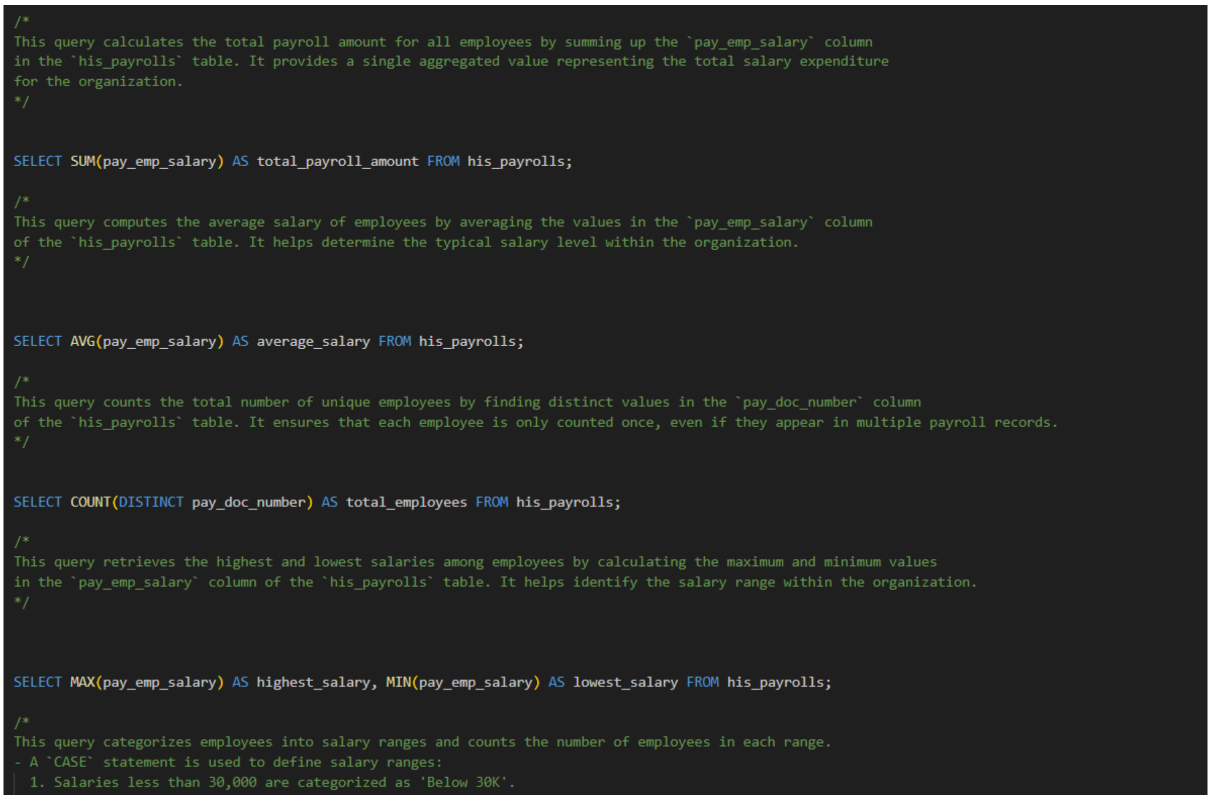
```sql
SELECT
    p.pat_id,
    p.pat_fname,
    p.pat_lname,
    p.pat_number,
    p.pat_addr,
    p.pat_phone,
    p.pat_age,
    p.pat_type,
    v.vit_bodytemp AS recent_body_temp,
    v.vit_heartpulse AS recent_heart_pulse,
    v.vit_resprate AS recent_resp_rate,
    v.vit_bloodpress AS recent_blood_pressure,
    pr.pres_date AS last_prescription_date,
    pr.pres_pat_ailment AS prescription_ailment,
    mdr.mdr_pat_ailment AS medical_record_ailment,
    mdr.mdr_date_rec AS medical_record_date
FROM
    his_patients AS p
LEFT JOIN
    his_vitals AS v ON p.pat_number = v.vit_pat_number
LEFT JOIN
    his_prescriptions AS pr ON p.pat_number = pr.pres_pat_number
LEFT JOIN
    his_medical_records AS mdr ON p.pat_number = mdr.mdr_pat_number
ORDER BY
    p.pat_fname, p.pat_lname;
```



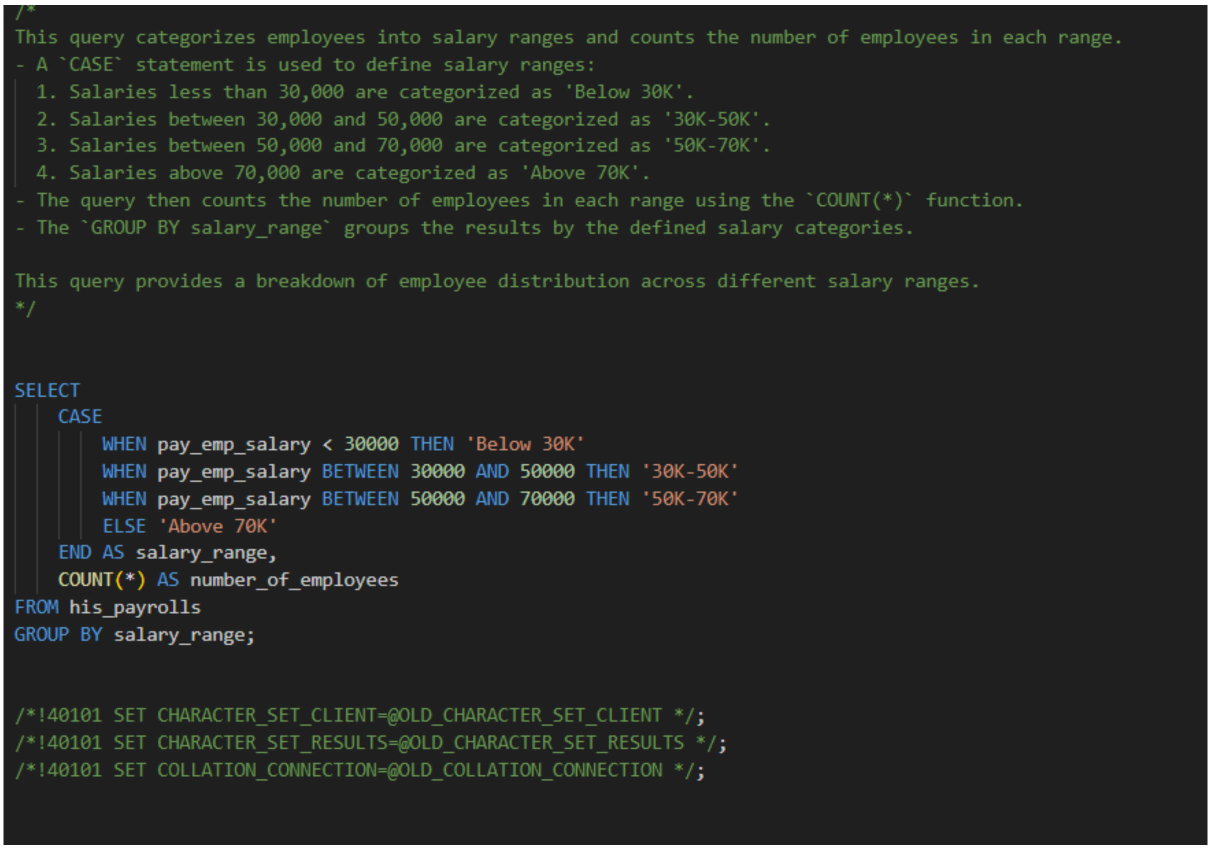
```sql
-- Retrieve all pharmaceutical details, ordered randomly

/*
This query retrieves pharmaceutical product details from the `his_pharmaceuticals` table and displays the following:
1. `Pharmaceutical Name`: Name of the product.
2. `Pharmaceutical Barcode`: Unique barcode identifying the product.
3. `Vendor ID`: The ID of the vendor supplying the pharmaceutical.
4. `Category`: The category or type of pharmaceutical product.
5. `Quantity`: Current stock quantity of the pharmaceutical.

The query applies a `RAND()` function in the `ORDER BY` clause to shuffle the order of results, providing a randomized display
of pharmaceutical products each time the query is run.
*/


SELECT
    phar_name AS 'Pharmaceutical Name',
    phar_bcode AS 'Pharmaceutical Barcode',
    phar_vendor AS 'Vendor ID',
    phar_cat AS 'Category',
    phar_qty AS 'Quantity'
FROM
    his_pharmaceuticals
ORDER BY
    RAND();
```



```sql
This query retrieves detailed information about pharmaceutical products from the `his_pharmaceuticals` table and enriches
it with vendor details from the `his_vendor` table. The key components are:

1. **Selected Columns**:
   - `Pharmaceutical Name`: Name of the pharmaceutical product.
   - `Pharmaceutical Barcode`: Unique barcode for identifying the product.
   - `Vendor Name`: The name of the vendor supplying the product, retrieved through a join with the `his_vendor` table.
   - `Category`: Type or category of the product.
   - `Quantity`: Stock level of the product.

2. **Join**:
   - The `JOIN` operation links `his_pharmaceuticals` with `his_vendor` using `phar_vendor` and `v_id` to retrieve the vendor's name.

3. **Filter Condition**:
   - Filters pharmaceuticals to include only those whose quantity (`phar_qty`) is greater than the average quantity of all pharmaceuticals.
   - The subquery `(SELECT AVG(phar_qty) FROM his_pharmaceuticals)` computes the average stock quantity.

This query is useful for identifying well-stocked pharmaceuticals and associating them with their vendors.
*/



SELECT
    p.phar_name AS 'Pharmaceutical Name',
    p.phar_bcode AS 'Pharmaceutical Barcode',
    v.v_name AS 'Vendor Name',
    p.phar_cat AS 'Category',
    p.phar_qty AS 'Quantity'
FROM
    his_pharmaceuticals AS p
JOIN
    his_vendor AS v ON p.phar_vendor = v.v_id
WHERE
    p.phar_qty > (
        SELECT
            AVG(phar_qty)
        FROM
            his_pharmaceuticals
    );


/*
```



```sql
/*
This query calculates the total payroll amount for all employees by summing up the `pay_emp_salary` column
in the `his_payrolls` table. It provides a single aggregated value representing the total salary expenditure
for the organization.
*/


SELECT SUM(pay_emp_salary) AS total_payroll_amount FROM his_payrolls;

/*
This query computes the average salary of employees by averaging the values in the `pay_emp_salary` column
of the `his_payrolls` table. It helps determine the typical salary level within the organization.
*/



SELECT AVG(pay_emp_salary) AS average_salary FROM his_payrolls;

/*
This query counts the total number of unique employees by finding distinct values in the `pay_doc_number` column
of the `his_payrolls` table. It ensures that each employee is only counted once, even if they appear in multiple payroll records.
*/


SELECT COUNT(DISTINCT pay_doc_number) AS total_employees FROM his_payrolls;

/*
This query retrieves the highest and lowest salaries among employees by calculating the maximum and minimum values
in the `pay_emp_salary` column of the `his_payrolls` table. It helps identify the salary range within the organization.
*/



SELECT MAX(pay_emp_salary) AS highest_salary, MIN(pay_emp_salary) AS lowest_salary FROM his_payrolls;

/*
This query categorizes employees into salary ranges and counts the number of employees in each range.
- A `CASE` statement is used to define salary ranges:
  1. Salaries less than 30,000 are categorized as 'Below 30K'.
```



```sql
/*
This query categorizes employees into salary ranges and counts the number of employees in each range.
- A `CASE` statement is used to define salary ranges:
  1. Salaries less than 30,000 are categorized as 'Below 30K'.
  2. Salaries between 30,000 and 50,000 are categorized as '30K-50K'.
  3. Salaries between 50,000 and 70,000 are categorized as '50K-70K'.
  4. Salaries above 70,000 are categorized as 'Above 70K'.
- The query then counts the number of employees in each range using the `COUNT(*)` function.
- The `GROUP BY salary_range` groups the results by the defined salary categories.

This query provides a breakdown of employee distribution across different salary ranges.
*/


SELECT
    CASE
        WHEN pay_emp_salary < 30000 THEN 'Below 30K'
        WHEN pay_emp_salary BETWEEN 30000 AND 50000 THEN '30K-50K'
        WHEN pay_emp_salary BETWEEN 50000 AND 70000 THEN '50K-70K'
        ELSE 'Above 70K'
    END AS salary_range,
    COUNT(*) AS number_of_employees
FROM his_payrolls
GROUP BY salary_range;


/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Link to the full project
https://github.com/TanayB10/hospital-management-system.git