Diagnosia - Database Schema Design

1. User Management Tables

users

Column	Type	Constraints	Description
user_id	SERIAL	PRIMARY KEY	Unique identifier
email	VARCHAR(255)	UNIQUE, NOT NULL	User email
password_hash	VARCHAR(255)	NOT NULL	Encrypted password
first_name	VARCHAR(100)	NOT NULL	First name
last_name	VARCHAR(100)	NOT NULL	Last name
phone	VARCHAR(20)	UNIQUE, NOT NULL	Phone number
date_of_birth	DATE	NOT NULL	Date of birth
gender	ENUM('male', 'female', 'other')	NOT NULL	Gender
is_active	BOOLEAN	DEFAULT TRUE	Account status
created_at	TIMESTAMP	DEFAULT NOW()	Registration timestamp
updated_at	TIMESTAMP	DEFAULT NOW()	Last update timestamp

user_roles

Column	Type	Constraints	Description
role_id	SERIAL	PRIMARY KEY	Unique role identifier
role_name	VARCHAR(50)	UNIQUE, NOT NULL	Role name
role_description	TEXT		Role description

Default Roles:

- `patient` - Book tests, view reports
- `sample_collector` - Manage sample collection
- `lab_technician` - Process samples, upload reports
- `admin` - Full system access
- `lab_manager` - Manage lab operations

User_role_assignments PRIMARY KEY (user_id, role_id)

Column	Type	Constraints	Description
user_id	INTEGER	FOREIGN KEY → users(user_id)	User reference
role_id	INTEGER	FOREIGN KEY → user_roles(role_id)	Role reference

user_addresses

Column	Type	Constraints	Description
address_id	SERIAL	PRIMARY KEY	Unique address identifier
user_id	INTEGER	FOREIGN KEY → users(user_id)	User reference
address	VARCHAR(500)	NOT NULL	Full address

2. Test Management Tables

test_categories

Column	Type	Constraints	Description
category_id	SERIAL	PRIMARY KEY	Unique category identifier
category_name	VARCHAR(100)	UNIQUE, NOT NULL	Category name
category_description	TEXT		Category description
category_icon	VARCHAR(255)		Icon URL/path

tests

Column	Type	Constraints	Description
test_code	VARCHAR(50)	PRIMARY KEY	Test code
test_name	VARCHAR(255)	NOT NULL	Test name
test_description	TEXT		Detailed description
category_id	INTEGER	FOREIGN KEY → test_categories(category_id)	Category reference
base_price	DECIMAL(10,2)	NOT NULL	Base test price
duration_hours	INTEGER	NOT NULL	Result processing time
preparation_instructions	TEXT		Pre-test instructions
sample_type	VARCHAR(100)	NOT NULL	Sample required (blood, urine, etc.)
fasting_required	BOOLEAN	DEFAULT FALSE	Fasting requirement
fasting_hours	INTEGER		Required fasting hours
gender_specific	VARCHAR(20)		Gender restriction if any

3. Appointment Management Tables

appointments

Column	Type	Constraints	Description
appointment_id	SERIAL	PRIMARY KEY	Unique appointment identifier
patient_id	INTEGER	FOREIGN KEY → users(user_id)	Patient reference
appointment_date	DATE	NOT NULL	Appointment date
appointment_time	TIME	NOT NULL	Appointment time
appointment_type	ENUM('lab_visit', 'home_collection')	NOT NULL	Appointment type
collection_address_id	INTEGER	FOREIGN KEY → user_addresses(address_id)	Collection address (if home)
status	ENUM('booked', 'cancelled', 'rescheduled', 'no_show', 'sample_collected', 'result_published')	DEFAULT 'booked'	Appointment status
total_amount	DECIMAL(10,2)	NOT NULL	Total appointment cost
special_instructions	TEXT		Special requirements
cancellation_reason	TEXT		Cancellation reason
cancelled_by	INTEGER	FOREIGN KEY → users(user_id)	Who cancelled
cancelled_at	TIMESTAMP		Cancellation timestamp
rescheduled_from	INTEGER	FOREIGN KEY → appointments(appointment_id)	Original appointment

rescheduled_reason	TEXT		Reschedule reason
created_at	TIMESTAMP	DEFAULT NOW()	Booking timestamp
updated_at	TIMESTAMP	DEFAULT NOW()	Last update timestamp

appointment_tests

Column	Type	Constraints	Description
appointment_test_id	SERIAL	PRIMARY KEY	Unique mapping identifier
appointment_id	INTEGER	FOREIGN KEY → appointments(appointment_id)	Appointment reference
test_code	VARCHAR(50)	FOREIGN KEY → tests(test_code)	Test reference
test_price	DECIMAL(10,2)	NOT NULL	Test price at booking
patient_name	VARCHAR(255)	NOT NULL	Patient name for test
patient_age	INTEGER	NOT NULL	Patient age
patient_gender	ENUM('male', 'female', 'other')	NOT NULL	Patient gender
status	ENUM('booked', 'cancelled', 'rescheduled', 'no_show', 'sample_collected', 'result_published')	DEFAULT 'booked'	Individual test status

4. Sample Management Tables

samples

Column	Type	Constraints	Description
sample_id	SERIAL	PRIMARY KEY	Unique sample identifier
sample_code	VARCHAR(50)	UNIQUE, NOT NULL	Sample barcode/ID
appointment_test_id	INTEGER	FOREIGN KEY → appointment_tests(appointment_test_id)	Test reference
collected_by	INTEGER	FOREIGN KEY → users(user_id)	Sample collector
collected_at	TIMESTAMP		Collection timestamp
collection_method	VARCHAR(100)		How sample was collected
sample_quality	ENUM('good', 'acceptable', 'poor', 'rejected')		Sample quality assessment
storage_location	VARCHAR(100)		Where sample is stored
temperature_maintained	BOOLEAN	DEFAULT TRUE	Temperature chain maintained
received_by_lab	INTEGER	FOREIGN KEY → users(user_id)	Lab technician who received
received_at	TIMESTAMP		Lab receipt timestamp
processing_started_at	TIMESTAMP		Processing start time
processing_completed_at	TIMESTAMP		Processing completion time

status	ENUM('collected', 'in_transit', 'received', 'processing', 'completed', 'rejected')	DEFAULT 'collected'	Sample status
rejection_reason	TEXT		Why sample was rejected
notes	TEXT		Additional notes

5. Results Management Tables

test_results

Column	Type	Constraints	Description
result_id	SERIAL	PRIMARY KEY	Unique result identifier
sample_id	INTEGER	FOREIGN KEY → samples(sample_id)	Sample reference
test_code	VARCHAR(50)	FOREIGN KEY → tests(test_code)	Test reference
processed_by	INTEGER	FOREIGN KEY → users(user_id)	Lab technician
verified_by	INTEGER	FOREIGN KEY → users(user_id)	Quality checker
result_values	JSONB	NOT NULL	Test results data
reference_ranges	JSONB		Normal reference ranges
abnormal_flags	JSONB		Abnormal value indicators
interpretation	TEXT		Result interpretation

recommendations	TEXT		Medical recommendations
critical_values	BOOLEAN	DEFAULT FALSE	Critical result flag
result_status	ENUM('draft', 'preliminary', 'final', 'amended', 'cancelled')	DEFAULT 'draft'	Result status
processed_at	TIMESTAMP	DEFAULT NOW()	Processing timestamp
verified_at	TIMESTAMP		Verification timestamp
released_at	TIMESTAMP		Release timestamp
amended_reason	TEXT		Amendment reason
notes	TEXT		Additional notes

test_reports

Column	Type	Constraints	Description
report_id	SERIAL	PRIMARY KEY	Unique report identifier
appointment_id	INTEGER	FOREIGN KEY → appointments(appointment_id)	Appointment reference
report_number	VARCHAR(50)	UNIQUE, NOT NULL	Report reference number
report_date	DATE	NOT NULL	Report generation date
report_type	ENUM('individual', 'consolidated')	NOT NULL	Report type

report_file_path	VARCHAR(500)		PDF file location
report_file_name	VARCHAR(255)		Original filename
file_size_kb	INTEGER		File size
generated_by	INTEGER	FOREIGN KEY → users(user_id)	Who generated report
approved_by	INTEGER	FOREIGN KEY → users(user_id)	Medical approval
patient_notified	BOOLEAN	DEFAULT FALSE	Notification sent flag
patient_downloaded	BOOLEAN	DEFAULT FALSE	Download status
download_count	INTEGER	DEFAULT 0	Download counter
first_downloaded_at	TIMESTAMP		First download time
last_downloaded_at	TIMESTAMP		Last download time
is_active	BOOLEAN	DEFAULT TRUE	Report validity
created_at	TIMESTAMP	DEFAULT NOW()	Creation timestamp

6. Payment Management Tables

payment_methods

Column	Type	Constraints	Description

method_id	SERIAL	PRIMARY KEY	Unique method identifier
method_name	VARCHAR(100)	NOT NULL	Payment method name
method_type	ENUM('online', 'cash', 'card', 'wallet', 'upi')	NOT NULL	Method type
is_active	BOOLEAN	DEFAULT TRUE	Method availability

payments

Column	Type	Constraints	Description
payment_id	SERIAL	PRIMARY KEY	Unique payment identifier
payment_reference	VARCHAR(100)	UNIQUE, NOT NULL	Payment reference
appointment_id	INTEGER	FOREIGN KEY → appointments(appointment_id)	Appointment reference
user_id	INTEGER	FOREIGN KEY → users(user_id)	User making payment
payment_method_id	INTEGER	FOREIGN KEY → payment_methods(method_id)	Payment method
amount	DECIMAL(10,2)	NOT NULL	Payment amount
currency	VARCHAR(10)	DEFAULT 'INR'	Currency code
payment_status	ENUM('success', 'failed', 'refunded')	DEFAULT 'success'	Payment status
gateway_response	JSONB		Payment gateway response

transaction_id	VARCHAR(100)		Gateway transaction ID
completed_at	TIMESTAMP	DEFAULT NOW()	Payment completion
failure_reason	TEXT		Failure description
refund_amount	DECIMAL(10,2)	DEFAULT 0	Refunded amount
refund_reason	TEXT		Refund reason
refunded_at	TIMESTAMP		Refund timestamp

7. Notification System Tables

notification_templates

Column	Type	Constraints	Description
template_id	SERIAL	PRIMARY KEY	Unique template identifier
template_name	VARCHAR(100)	UNIQUE, NOT NULL	Template name
event_trigger	VARCHAR(100)	NOT NULL	Triggering event
subject_template	VARCHAR(500)		Email subject template
body_template	TEXT	NOT NULL	Email body template
variables	JSONB		Available variables
is_active	BOOLEAN	DEFAULT TRUE	Template status
created_at	TIMESTAMP	DEFAULT NOW()	Creation timestamp

notifications

Column	Type	Constraints	Description

notification_id	SERIAL	PRIMARY KEY	Unique notification identifier
user_id	INTEGER	FOREIGN KEY → users(user_id)	Recipient
template_id	INTEGER	FOREIGN KEY → notification_templates(template_id)	Template used
subject	VARCHAR(500)		Email subject
message	TEXT	NOT NULL	Email message
status	ENUM('pending', 'sent', 'delivered', 'failed')	DEFAULT 'pending'	Email status
priority	ENUM('low', 'normal', 'high', 'urgent')	DEFAULT 'normal'	Priority level
related_entity_type	VARCHAR(50)		Related entity (appointment, result, etc.)
related_entity_id	INTEGER		Related entity ID
scheduled_at	TIMESTAMP		Scheduled send time
sent_at	TIMESTAMP		Actual send time
delivered_at	TIMESTAMP		Delivery confirmation
failure_reason	TEXT		Failure description
retry_count	INTEGER	DEFAULT 0	Retry attempts

created_at	TIMESTAMP	DEFAULT NOW()	Creation timestamp

8. System Configuration Tables

system_settings

Column	Type	Constraints	Description
setting_id	SERIAL	PRIMARY KEY	Unique setting identifier
setting_key	VARCHAR(100)	UNIQUE, NOT NULL	Setting key
setting_value	TEXT		Setting value
setting_type	ENUM('string', 'number', 'boolean', 'json')	DEFAULT 'string'	Value type
description	TEXT		Setting description
category	VARCHAR(100)		Setting category
is_public	BOOLEAN	DEFAULT FALSE	Publicly accessible
updated_by	INTEGER	FOREIGN KEY → users(user_id)	Who updated
updated_at	TIMESTAMP	DEFAULT NOW()	Update timestamp

audit_logs

Column	Type	Constraints	Description
log_id	SERIAL	PRIMARY KEY	Unique log identifier
user_id	INTEGER	FOREIGN KEY → users(user_id)	User performing action
action	VARCHAR(100)	NOT NULL	Action performed
entity_type	VARCHAR(100)	NOT NULL	Affected entity type

entity_id	INTEGER		Affected entity ID
old_values	JSONB		Previous values
new_values	JSONB		New values
ip_address	INET		User IP address
user_agent	TEXT		User browser/app
timestamp	TIMESTAMP	DEFAULT NOW()	Action timestamp
session_id	VARCHAR(255)		User session

10. Chatbot Support Tables

chatbot_conversations

Column	Type	Constraints	Description
conversation_id	SERIAL	PRIMARY KEY	Unique conversation identifier
user_id	INTEGER	FOREIGN KEY → users(user_id)	User (if logged in)
session_id	VARCHAR(255)	NOT NULL	Session identifier
started_at	TIMESTAMP	DEFAULT NOW()	Conversation start
ended_at	TIMESTAMP		Conversation end
status	ENUM('active', 'ended', 'escalated')	DEFAULT 'active'	Conversation status
escalated_to	INTEGER	FOREIGN KEY → users(user_id)	Human agent
satisfaction_rating	INTEGER		User rating (1-5)
feedback	TEXT		User feedback

chatbot_messages

Column	Type	Constraints	Description
message_id	SERIAL	PRIMARY KEY	Unique message identifier
conversation_id	INTEGER	FOREIGN KEY → chatbot_conversations(conversation_id)	Conversation reference
sender_type	ENUM('user', 'bot', 'agent')	NOT NULL	Message sender
message_text	TEXT	NOT NULL	Message content
intent	VARCHAR(100)		Detected intent
confidence_score	DECIMAL(5,4)		Confidence level
response_time_ms	INTEGER		Bot response time
created_at	TIMESTAMP	DEFAULT NOW()	Message timestamp

Database Relationships Summary

Primary Relationships:

1. **Users → Roles (Many-to-Many)**
    - `user_role_assignments` links users to roles
2. **Users → Addresses (One-to-Many)**
    - Users can have multiple addresses
3. **Tests → Categories (Many-to-One)**
    - Each test belongs to one category
4. **Appointments → Users (Many-to-One)**
    - Each appointment belongs to one patient
5. **Appointments → Tests (Many-to-Many)**
    - `appointment_tests` links appointments to tests
6. **Samples → Appointment Tests (One-to-One)**
    - Each sample corresponds to one appointment test
7. **Test Results → Samples (One-to-One)**
    - Each sample has one result
8. **Payments → Appointments (Many-to-One)**
    - Multiple payments can be made for one appointment
9. **Lab Centers → Tests (Many-to-Many)**
    - `center_tests` links centers to available tests
10. **Staff Assignments → Appointments (Many-to-Many)**
    - `appointment_assignments` tracks staff assignments