HealthLink

Tommy Liu - tommy.liu@sjsu.edu
Fnu Hasham - fnu.hasham@sjsu.edu
Hiba Hasan - hiba.hasan@sjsu.edu

Team 3

San Jose State University

CS157A, Sec 01

Dr. Mike Wu

July 5th, 2025

Table of Contents

Functional Requirements

The application will provide functionality for the active user role. A user can have either the role of patient or doctor. A patient and a doctor can communicate with each other directly, without contacting each other in-person-this is the main feature of the app. A patient will be able to record their health information in a convenient platform, such as managing medications, health problems, vaccination history, etc. A patient can also view their medical data, as reported by their doctors, over a period of time. As such, doctors will be able to record and report medical data for their patients. Doctors will be able to respond to their patients directly, providing instant feedback to their questions and comments. Doctors can prescribe their patients new medications, adjust dosages, and also be able to keep notes for their patients. Finally, a separate Admin user will be available for maintenance and testing purposes.

Functions

1. Patients can register and log in securely.
- The patient can securely register a new account
- The patient can securely log in and be able to access their information.

I/O:

To login:

1. The user enters their username and password into the form.
2. The system verifies their credentials and redirects them to the dashboard if authenticated. Otherwise, they are notified that login failed.

To register:

1. The user enters their first and last name, email address, password, and selects the Patient into the form.
2. The system reads the form fields, and creates a new user tuple.

2. Doctors can securely register, log in, and view assigned patients.
- The doctor can register an account
- The doctor can log in and view patients assigned to them and perform various actions on them

I/O:

To login:

3. The user enters their username and password into the form.
4. The system verifies their credentials and redirects them to the dashboard if authenticated. Otherwise, they are notified that login failed.

To register:

3. The user enters their first and last name, email address, password, and selects the role as Doctor into the form.
4. The system reads the form fields, and creates a new user tuple.

3. Patients can document their health and well-being, and symptoms.
- The patient can document any of their symptoms, if any
- The patient can make a note of any other useful information for their doctor, such as questions or concerns they might have
- The system will then securely save this information in the database.

I/O:

- The user visits the page to create a note or clicks the quick note button.
- The user documents via typing in the text field and the user submits the form.
- Their text content is used to create a new tuple in the note table.

4. Patients can log medications and dosages.
- The patient can document when they take a medication, start taking a new medication, and their respective dosages.
- This helps the doctor make better and more informed decisions for their patients.
- The system will keep a record of this and save to the database.

I/O:

- The user visits the page to log their medications.
- The user fills out the form, selecting which medication (from the list assigned to them), the dose taken, and the time taken.
- The form fields are used to create a new tuple in the medication_log table.

5. Patients can upload test results (PDF/image format).
- The patient can document their medical test reports
- This helps the doctor make better and more informed decisions for their patients.
- The system will keep a record of this and save it to the database.

I/O:

- The user visits the page to upload sensitive documents.
- The user fills out the form, uploading the documents (image/pdf).
- The system will keep track of the uploaded documents and create a new tuple for it, storing it as a BLOB.

6. Doctors can view patients’ logs in a dashboard.
- The doctor can view historical information about their patients and track their health over time
- This helps the doctor make better and more informed decisions for their patients.

I/O:

- The doctor visits the dashboard page.
- The doctor finds the patient they want to view logs for.
- The page redirects to a subroute targeted for this patient, which displays their logs.

7. Doctors can provide comments or advice based on logs.
- The doctor can directly communicate with the patients, provide comments, and address concerns

I/O:

- The doctor visits a chatpage route associated with that patient.
- The doctor can type a message to communicate remotely with that patient.

8. Doctors can make notes for their patients.
- The doctor can make notes for a patient, allowing them to maintain an active history for them.
- The system will be to deliver the doctor’s note(s) for a patient. The page will be able to display all the notes the doctor has on the patient, and will be able to create a new note or update an existing note.

I/O:

- The doctor visits the dashboard page associated with that patient.
- The doctor fills out the form, such as the note content. The database will keep track of this information.

9. Role-based access control (patients can't view other patients, etc.).
- Roles can be a patient or a doctor
- Patients cannot access sensitive information about each other
- To protect patients’ personal information, ensuring privacy and system security
- The system must perform checks on the role of the user and whether they are authorized to perform the action.

I/O:

- The system keeps track of the active user, thereby preventing unauthorized accesses to other users content.
- The user must first be authenticated by entering their email addrses and password. Unauthenticated users are blocked from interacting with sensitive information.

10. Admin can manage users and monitor activity.
- To ensure system security, reliability, and performance.
- The system will keep relevant logs for maintenance purposes, quality assurance, or for metrics

I/O:
- The admin visits a dedicated logs page that details system events.

11. A notification system for doctor feedback or missed log entries.
- Brief update system to inform patients if their doctor made a note regarding their medications, health, etc.
- The system will keep track of database changes relevant to the patient, and check if there were any updates
- The page will display a small toast if the doctor has responded to their questions, made adjustments to medications, etc.

I/O:
- Toast system to display recent actions taken by the doctor.

12. Patients can view their full medical history, including past symptom logs, medication, and test results.
- The system will keep track of their medical history (including past symptoms, medications, and test results)
- The page will have graphical displays of various health data points so they can look for any trends

I/O:
- The patient visits their report page, which shows a quick summary of their medical report.

13. Doctors can update and manage treatment plans for each patient.
- Doctors can search for the patient to manage, which could have a filter based on their characteristics
- Doctors can create/update/delete patient medication dosages
- The system will keep track of patient medications and their dosage. The page will display a list of the patient's medications and allow the patient’s doctor to assign, update, or delete a medication for them.

I/O:
- The doctor visits the dashboard page for that patient, going to their Treatment Plan page.

- The doctor can manage the current treatment plans, or make any adjustments.

14. All medical history data is stored securely and accessible only to the respective patient and their doctors.
- Patients are only able to see pages relevant to them, with their health data
- Doctors cannot access patient data from other doctors

Non-Functional Issues

Graphical User Interface (GUI):

The application will have a clean and user-friendly interface, comprised of HTML, CSS, and JavaScript; it must be highly functional and practical. There will be a landing login page/account registration page. Based on the user credentials submitted in the form, it will direct the user to the portal, which respects the user's role. There will be a sidebar that acts as the main method of navigating throughout the application, containing quick and direct links to app features. Certain elements, such as input fields, forms, and buttons, are strictly required for user interactions. Some examples of pages will be for patient information, patient medical data and trends, and patient notes and records. Since medical data is tracked, we could create graphical displays for visual representation of trends and changes in a patient's health.

Security:

For an application where user information is highly sensitive, we must ensure the highest levels of user security and privacy. Therefore, all user data will be encrypted before being saved in the RDBMS. Passwords will be hashed and not be stored in plaintext. The app will implement protection against SQL injection attacks and unauthorized access attempts. Since user input is the main source of data, we must ensure that the system cannot be compromised and that all inputs must be validated and sanitized before reaching the database.

Patient and doctor names will be visible throughout the app to ensure patients are speaking with their assigned doctor and doctors are working with their assigned patients. Patients will go through email verification through a trusted email provider platform when registering, as well as ensuring password complexity to avoid the various password cracking attacks, such as dictionary, brute-force, or denial of service attacks. We can also rate limit login attempts, as well as provide a way for users to reset their login credentials.

Access Control:

The system will utilize role-based access control (RBAC). RBAC helps to make sure users can perform the functions assigned to their role only. So, patients can only view and edit their records, and doctors can only access the records of patients assigned to them. Admins can manage user roles and monitor usage logs.

Performance & Scalability:

The app will be optimized to handle multiple concurrent users without performance lags or system instability. For example, there may be situations where a doctor requires immediate access to patient records to appropriately respond to a medical situation.

Reliability

The application should be able to handle errors gracefully, such as failed database connections or invalid inputs from users. In the event of invalid input, the system should be able to inform the user and to help them send valid data. Admins should be notified of system errors, such as failed database errors or runtime exceptions. Error messages will be user-friendly and not expose technical details whatsoever.

Data Privacy

Given the sensitive nature of patient health records, the app will adhere to basic data privacy principles. This way, patient information is treated with the highest level of confidentiality. We will ensure that only authorized users will have access to sensitive data. As earlier mentioned, patients should only be able to create, read, and modify their own information and have no interaction with other patient records. Doctors can only create, read, and modify records of patients assigned to them.

Entities, Attributes, Relationships with Multiplicity

1. User
    - user_id
    - email_address
    - password_hashed
    - first_name
    - last_name
    - role (patient, doctor, admin)
    - created_at
    - updated_at

    Entity User represents a User on the web application (authentication and authorization). A User consists of user_id as the primary key, their role (patient, doctor, or admin), first and last name, email address and hashed password. user_id will be used as foreign key for Patient and Doctor-related entities. There is a one-one relationship from user-patient and user-doctor. There is one-many relationship between user-system_logs.

2. Patient
   - date_of_birth
   - phone_number
   - address
   - emergency_contact_name
   - emergency_contact_phone_number

   Entity Patient represents a Patient for some doctor at this health service provider. A Patient is a subclass of User, but has new properties such as their date of birth, phone number, address, emergency contact name and phone number. A Patient's user_id will be used as foreign key for entities such as Medications, Doctor-Patient relation, and Test Results. There is one-many relationship from patient-medication, patient-medication_log, patient-test_result, patient-note, and patient-notification. There is many-many relationship from doctor-patient.

3. Doctor
   - department

   Entity Doctor represents a Doctor for this health service provider. A doctor is a subclass of User, so to distinguish between the two, a doctor has a department attribute. There is a one-many relationship from doctor-medications. There is many-many relationship from doctor-patient.

4. Assigned_to
   - assignment_id
   - doctor_id
   - patient_id

   The relation between a Doctor entity and Patient entity. The assignment_id is auto generated as the primary key, and doctor_id and patient_id as foreign keys pointing to the doctor assigned to the patient and patient assigned to the doctor, respectively.

5. Medication
   - medication_id
   - patient_id
   - doctor_id
   - name
   - dosage

- frequency
- notes

Entity Medication is for tracking medications that are assigned to a patient by a doctor. We have foreign keys patient_id and doctor_id to know who the medication is for and who assigned the medication. We also keep track of other information about the medication, such as its name, the dosage, frequency, etc.

6. Medication Log

- medication_log_id
- medication_id
- patient_id
- dosage_taken
- note
- taken_at
- created_at

Entity Medication Log is for tracking when a Patient takes their prescribed medication. We have foreign keys to medication_id and patient_id so we can associate which Patient is taking which Medication. We also keep track of how much of their dose they took, the time, and other information they might have. Patient can have one-many relationship.

7. System Log

- log_id
- user_id
- action
- detail
- timestamp

Entity System Log tracks system activity for security and auditing. The primary key is log_id. It contains user_id as a foreign key to identify who performed the action. Other attributes include the action type, detailed description of the event, and the timestamp when it occurred. This helps maintain transparency and accountability in the system. User can have one-many system logs entries.

8. Test Result

- test_id
- patient_id
- uploaded_by
- file_path

- test_type
- uploaded_at

Entity TestResult stores medical test files related to a patient. The primary key is test_id. It includes patient_id and uploaded_by as foreign keys patient_id links to the patient the test belongs to, and uploaded_by identifies who uploaded the result (could be a doctor or patient). Additional attributes include the file path, test type, and a timestamp for upload time. Patient can have one-many test-result.

9. Messages

- message_id
- sender_id
- receiver_id
- content
- timestamp

Entity Message is for direct communication between a patient and a doctor. The primary key is message_id, which uniquely identifies each message. We store sender_id and receiver_id as foreign keys to track who sent the message and who received it. Each message also includes the content and a timestamp to show when it was sent. Patient/Doctor can have one-many Messages.

10.Notifications

- notification_id
- user_id
- message
- created_at

Entity Notification is used to alert a Patient when their Doctor provides feedback, updates a medication, or adds a note. A notification can also be available for a Doctor, such as a new message from a patient. The primary key is notification_id. It includes user_id as a foreign key to identify who the notification is for. Other attributes include the message content, a timestamp showing when the notification was created, and a read status to track whether the patient has seen it. Patient can have zero-to-many Notifications.

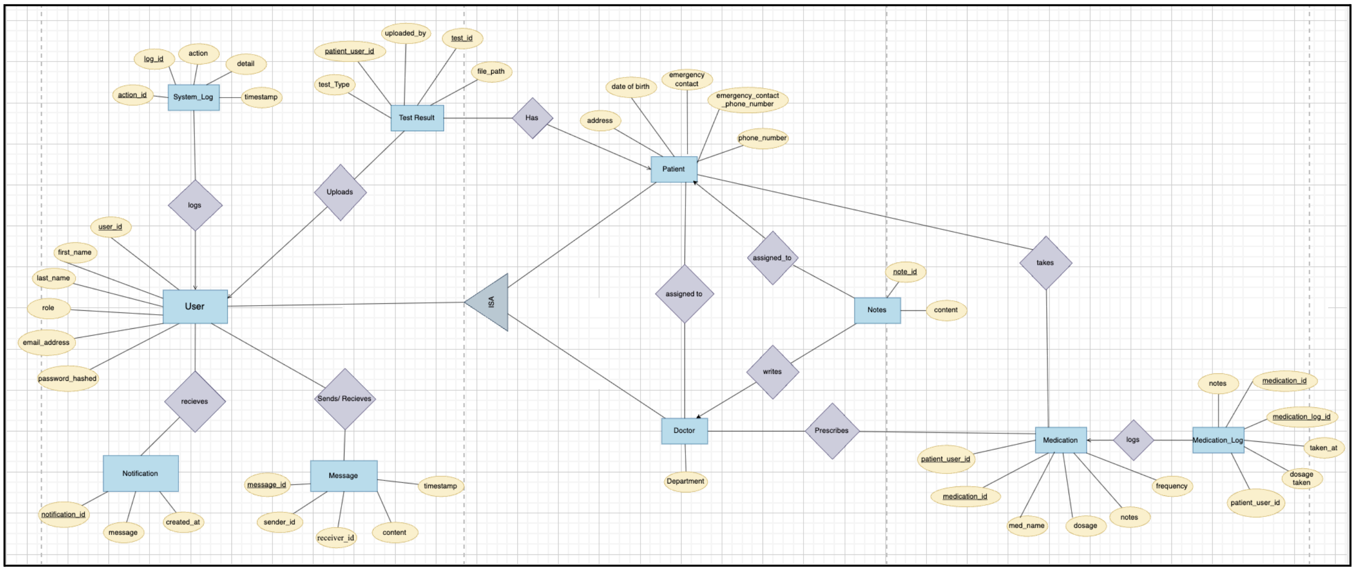
11.Note

- note_id
- patient_id
- doctor_id
- content

- created_at

Entity Note represents a note written by a doctor for a specific patient, usually containing treatment feedback, symptom evaluations, or personalized instructions. Similarly, a Patient can write a Note for a specific doctor, or it could be some general note for themselves (NULL doctor_id). Each note is uniquely identified by note_id, and could one doctor_id to one patient_id. User can have zero-to-many messages.

Entity-Relationship Diagram (ERD)



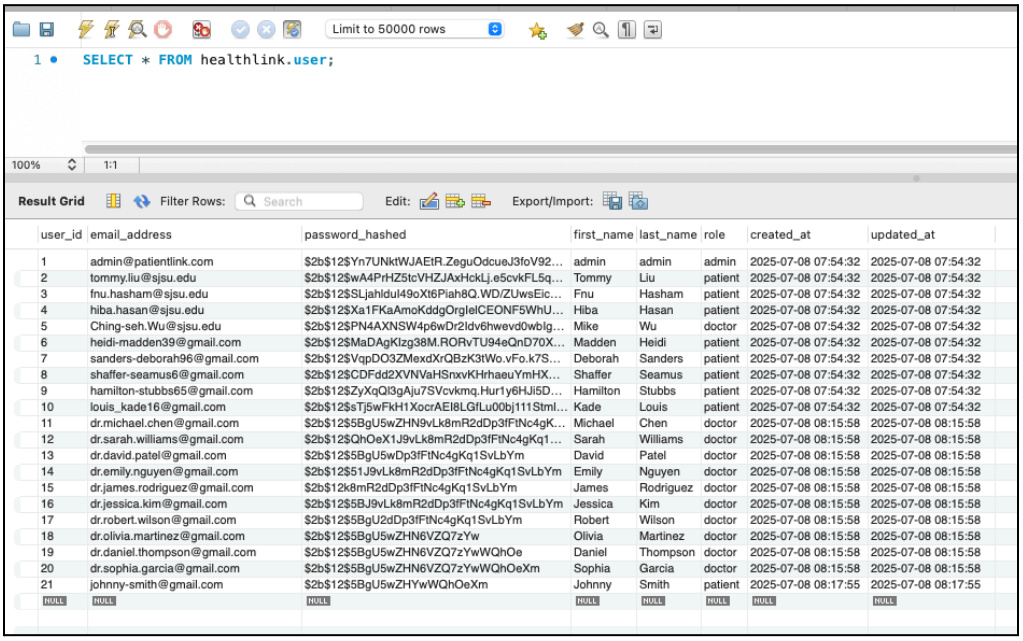
ERD Schemas

- User(user_id, email_address, password_hashed, first_name, last_name, role, created_at, updated_at)
- Patient(patient_id, date_of_birth, phone_number, address, emergency_contact_name, emergency_contact_phone_number)
- Doctor(doctor_id, department)
- Assigned_to(assigned_id, patient_id, doctor_id)
- Medication(medication_id, patient_id, doctor_id, name, dosage, frequency, notes)
- Medication_log(medication_log_id, medication_id, patient_id, taken_at, dosage_taken, created_at, note)
- System_log(log_id, user_id, action, detail, timestamp)
- Test_result(test_id, patient_id, uploaded_by, file_path, test_type, uploaded_at)
- Message(message_id, sender_id, receiver_id, content, timestamp)
- Notification(notification_id, user_id, message, created_at)

- Note(note_id, doctor_id, patient_id, content, created_at)

MySQL Workbench Table Contents

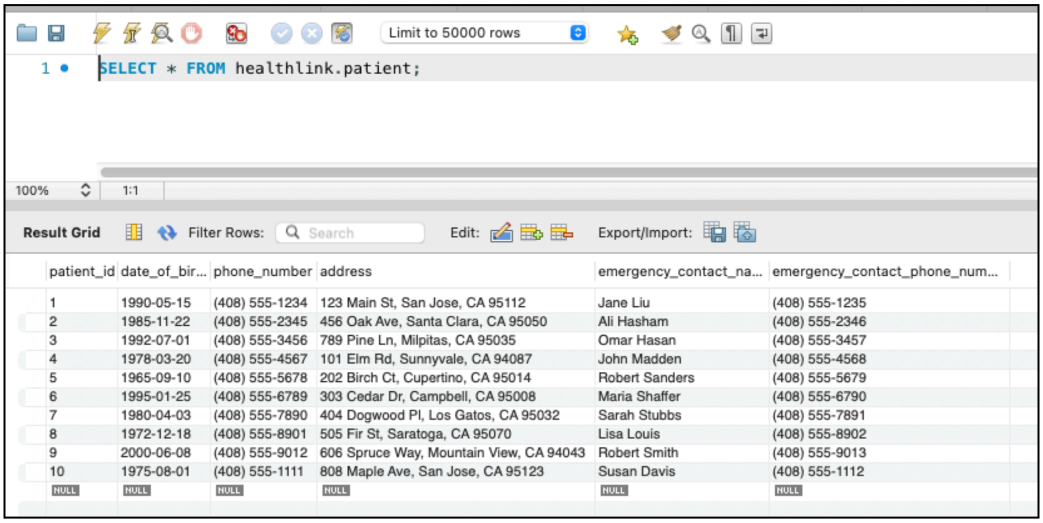
1. User(user_id, email_address, password_hashed, first_name, last_name, role, created_at, updated_at)



```
1 • SELECT * FROM healthlink.user;
```

user_id	email_address	password_hashed	first_name	last_name	role	created_at	updated_at
1	admin@patientlink.com	$2b$12$Yn7UNktWJAEtR.ZeguOdcueJ3foV92…	admin	admin	admin	2025-07-08 07:54:32	2025-07-08 07:54:32
2	tommy.liu@sjsu.edu	$2b$12$wA4PrHZ5tcVHZJAxHckLj.e5cvkFL5q…	Tommy	Liu	patient	2025-07-08 07:54:32	2025-07-08 07:54:32
3	fnu.hasham@sjsu.edu	$2b$12$SLjahldul49oXt6Piah8Q.WD/ZUwsEic…	Fnu	Hasham	patient	2025-07-08 07:54:32	2025-07-08 07:54:32
4	hiba.hasan@sjsu.edu	$2b$12$Xa1FKaAmoKddgOrgIelCEONF5WhU…	Hiba	Hasan	patient	2025-07-08 07:54:32	2025-07-08 07:54:32
5	Ching-seh.Wu@sjsu.edu	$2b$12$PN4AXNSW4p6wDr2Idv6hwevd0wblg…	Mike	Wu	doctor	2025-07-08 07:54:32	2025-07-08 07:54:32
6	heidi-madden39@gmail.com	$2b$12$MaDAgKlzg38M.RORvTU94eQnD70X…	Madden	Heidi	patient	2025-07-08 07:54:32	2025-07-08 07:54:32
7	sanders-deborah96@gmail.com	$2b$12$VqpDO3ZMexdXrQBzK3tWo.vFo.k7S…	Deborah	Sanders	patient	2025-07-08 07:54:32	2025-07-08 07:54:32
8	shaffer-seamus6@gmail.com	$2b$12$CDFdd2XVNVaHSnxvKHrhaeuYmHX…	Shaffer	Seamus	patient	2025-07-08 07:54:32	2025-07-08 07:54:32
9	hamilton-stubbs65@gmail.com	$2b$12$ZyXqQl3gAju7SVcvkmq.Hur1y6HJi5D…	Hamilton	Stubbs	patient	2025-07-08 07:54:32	2025-07-08 07:54:32
10	louis_kade16@gmail.com	$2b$12$sTj5wFkH1XocrAEI8LGfLu00bj111Stml…	Kade	Louis	patient	2025-07-08 07:54:32	2025-07-08 07:54:32
11	dr.michael.chen@gmail.com	$2b$12$5BgU5wZHN9vLk8mR2dDp3fFtNc4gK…	Michael	Chen	doctor	2025-07-08 08:15:58	2025-07-08 08:15:58
12	dr.sarah.williams@gmail.com	$2b$12$QhOeX1J9vLk8mR2dDp3fFtNc4gKq1…	Sarah	Williams	doctor	2025-07-08 08:15:58	2025-07-08 08:15:58
13	dr.david.patel@gmail.com	$2b$12$5BgU5wDp3fFtNc4gKq1SvLbYm	David	Patel	doctor	2025-07-08 08:15:58	2025-07-08 08:15:58
14	dr.emily.nguyen@gmail.com	$2b$12$51J9vLk8mR2dDp3fFtNc4gKq1SvLbYm	Emily	Nguyen	doctor	2025-07-08 08:15:58	2025-07-08 08:15:58
15	dr.james.rodriguez@gmail.com	$2b$12k8mR2dDp3fFtNc4gKq1SvLbYm	James	Rodriguez	doctor	2025-07-08 08:15:58	2025-07-08 08:15:58
16	dr.jessica.kim@gmail.com	$2b$12$5BJ9vLk8mR2dDp3fFtNc4gKq1SvLbYm	Jessica	Kim	doctor	2025-07-08 08:15:58	2025-07-08 08:15:58
17	dr.robert.wilson@gmail.com	$2b$12$5BgU2dDp3fFtNc4gKq1SvLbYm	Robert	Wilson	doctor	2025-07-08 08:15:58	2025-07-08 08:15:58
18	dr.olivia.martinez@gmail.com	$2b$12$5BgU5wZHN6VZQ7zYw	Olivia	Martinez	doctor	2025-07-08 08:15:58	2025-07-08 08:15:58
19	dr.daniel.thompson@gmail.com	$2b$12$5BgU5wZHN6VZQ7zYwWQhOe	Daniel	Thompson	doctor	2025-07-08 08:15:58	2025-07-08 08:15:58
20	dr.sophia.garcia@gmail.com	$2b$12$5BgU5wZHN6VZQ7zYwWQhOeXm	Sophia	Garcia	doctor	2025-07-08 08:15:58	2025-07-08 08:15:58
21	johnny-smith@gmail.com	$2b$12$5BgU5wZHYwWQhOeXm	Johnny	Smith	patient	2025-07-08 08:17:55	2025-07-08 08:17:55
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

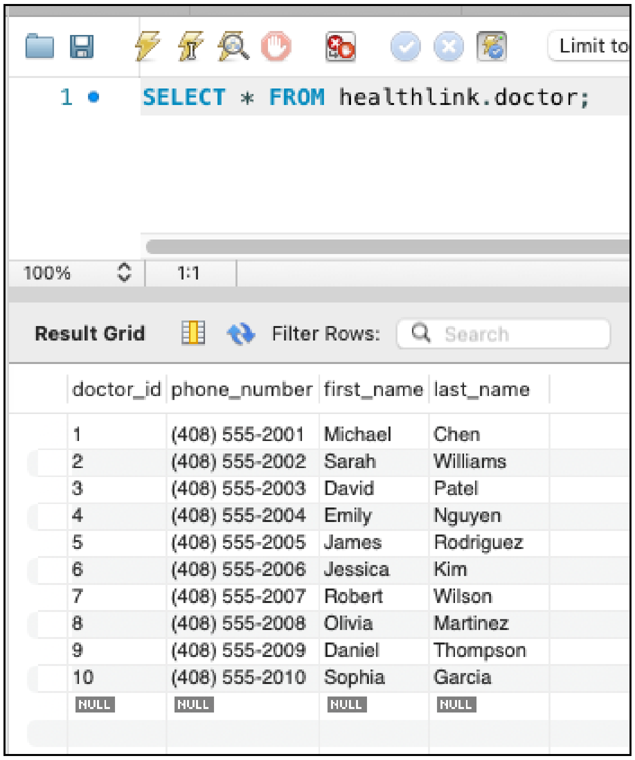
2. Patient(patient_id, date_of_birth, phone_number, address, emergency_contact_name, emergency_contact_phone_number)



```
1 • SELECT * FROM healthlink.patient;
```

patient_id	date_of_bir...	phone_number	address	emergency_contact_na...	emergency_contact_phone_num...
1	1990-05-15	(408) 555-1234	123 Main St, San Jose, CA 95112	Jane Liu	(408) 555-1235
2	1985-11-22	(408) 555-2345	456 Oak Ave, Santa Clara, CA 95050	Ali Hasham	(408) 555-2346
3	1992-07-01	(408) 555-3456	789 Pine Ln, Milpitas, CA 95035	Omar Hasan	(408) 555-3457
4	1978-03-20	(408) 555-4567	101 Elm Rd, Sunnyvale, CA 94087	John Madden	(408) 555-4568
5	1965-09-10	(408) 555-5678	202 Birch Ct, Cupertino, CA 95014	Robert Sanders	(408) 555-5679
6	1995-01-25	(408) 555-6789	303 Cedar Dr, Campbell, CA 95008	Maria Shaffer	(408) 555-6790
7	1980-04-03	(408) 555-7890	404 Dogwood Pl, Los Gatos, CA 95032	Sarah Stubbs	(408) 555-7891
8	1972-12-18	(408) 555-8901	505 Fir St, Saratoga, CA 95070	Lisa Louis	(408) 555-8902
9	2000-06-08	(408) 555-9012	606 Spruce Way, Mountain View, CA 94043	Robert Smith	(408) 555-9013
10	1975-08-01	(408) 555-1111	808 Maple Ave, San Jose, CA 95123	Susan Davis	(408) 555-1112
NULL	NULL	NULL	NULL	NULL	NULL

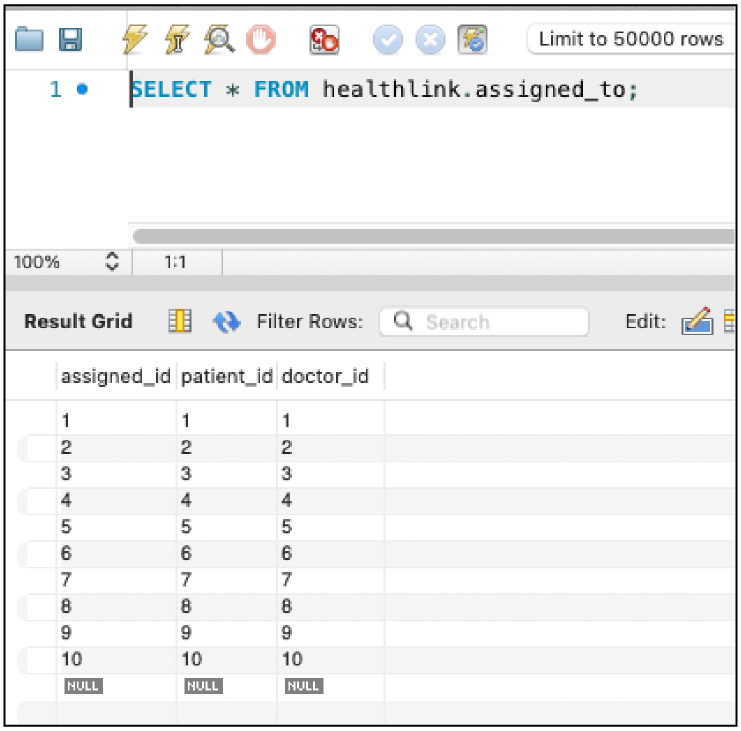
3. Doctor(doctor_id, phone_number, first_name, last_name)



```sql
SELECT * FROM healthlink.doctor;
```

doctor_id	phone_number	first_name	last_name
1	(408) 555-2001	Michael	Chen
2	(408) 555-2002	Sarah	Williams
3	(408) 555-2003	David	Patel
4	(408) 555-2004	Emily	Nguyen
5	(408) 555-2005	James	Rodriguez
6	(408) 555-2006	Jessica	Kim
7	(408) 555-2007	Robert	Wilson
8	(408) 555-2008	Olivia	Martinez
9	(408) 555-2009	Daniel	Thompson
10	(408) 555-2010	Sophia	Garcia
NULL	NULL	NULL	NULL

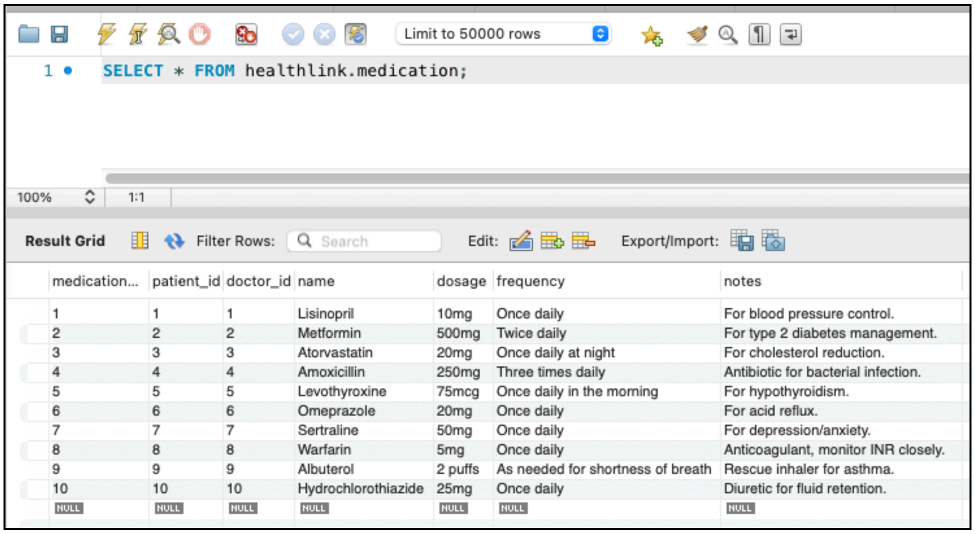
4. Assigned_to(assigned_id, patient_id, doctor_id)



```sql
SELECT * FROM healthlink.assigned_to;
```

assigned_id	patient_id	doctor_id
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
NULL	NULL	NULL

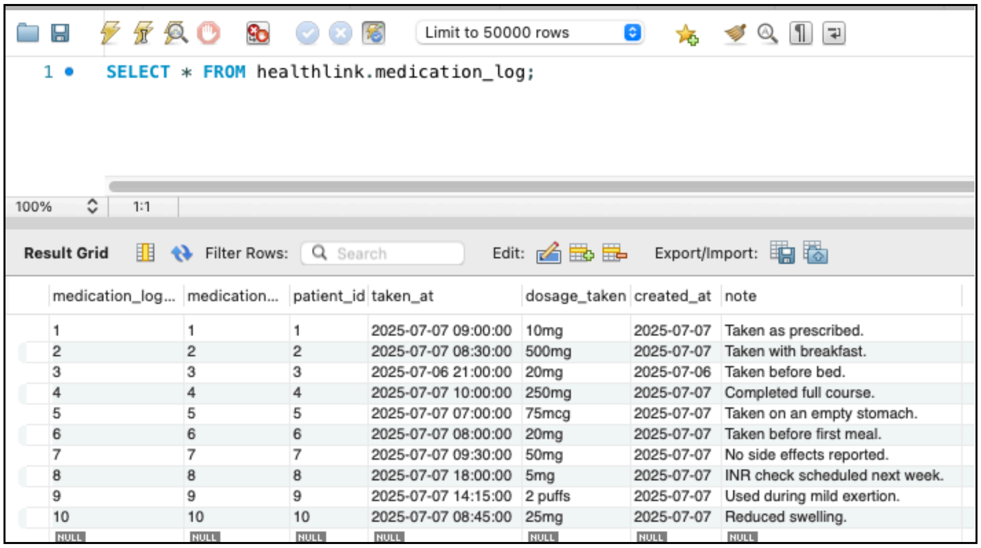
5. Medication(medication_id, patient_id, doctor_id, name, dosage, frequency, notes)



```
SELECT * FROM healthlink.medication;
```

medication...	patient_id	doctor_id	name	dosage	frequency	notes
1	1	1	Lisinopril	10mg	Once daily	For blood pressure control.
2	2	2	Metformin	500mg	Twice daily	For type 2 diabetes management.
3	3	3	Atorvastatin	20mg	Once daily at night	For cholesterol reduction.
4	4	4	Amoxicillin	250mg	Three times daily	Antibiotic for bacterial infection.
5	5	5	Levothyroxine	75mcg	Once daily in the morning	For hypothyroidism.
6	6	6	Omeprazole	20mg	Once daily	For acid reflux.
7	7	7	Sertraline	50mg	Once daily	For depression/anxiety.
8	8	8	Warfarin	5mg	Once daily	Anticoagulant, monitor INR closely.
9	9	9	Albuterol	2 puffs	As needed for shortness of breath	Rescue inhaler for asthma.
10	10	10	Hydrochlorothiazide	25mg	Once daily	Diuretic for fluid retention.
NULL	NULL	NULL	NULL	NULL	NULL	NULL

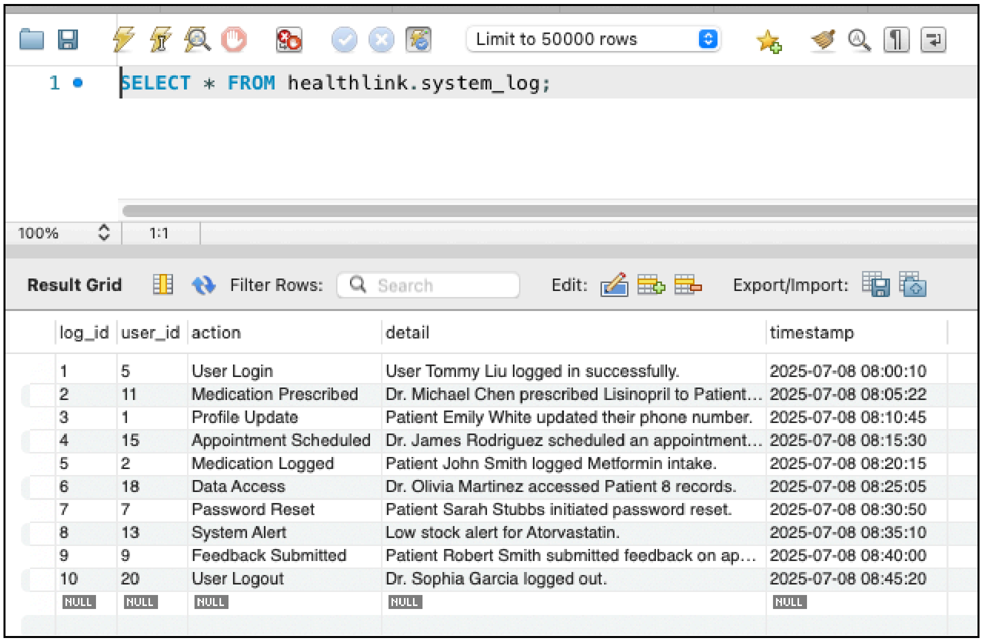
6. Medication_log(medication_log_id, medication_id, patient_id, taken_at, dosage_taken, created_at, note)



```
SELECT * FROM healthlink.medication_log;
```

medication_log...	medication...	patient_id	taken_at	dosage_taken	created_at	note
1	1	1	2025-07-07 09:00:00	10mg	2025-07-07	Taken as prescribed.
2	2	2	2025-07-07 08:30:00	500mg	2025-07-07	Taken with breakfast.
3	3	3	2025-07-06 21:00:00	20mg	2025-07-06	Taken before bed.
4	4	4	2025-07-07 10:00:00	250mg	2025-07-07	Completed full course.
5	5	5	2025-07-07 07:00:00	75mcg	2025-07-07	Taken on an empty stomach.
6	6	6	2025-07-07 08:00:00	20mg	2025-07-07	Taken before first meal.
7	7	7	2025-07-07 09:30:00	50mg	2025-07-07	No side effects reported.
8	8	8	2025-07-07 18:00:00	5mg	2025-07-07	INR check scheduled next week.
9	9	9	2025-07-07 14:15:00	2 puffs	2025-07-07	Used during mild exertion.
10	10	10	2025-07-07 08:45:00	25mg	2025-07-07	Reduced swelling.
NULL	NULL	NULL	NULL	NULL	NULL	NULL

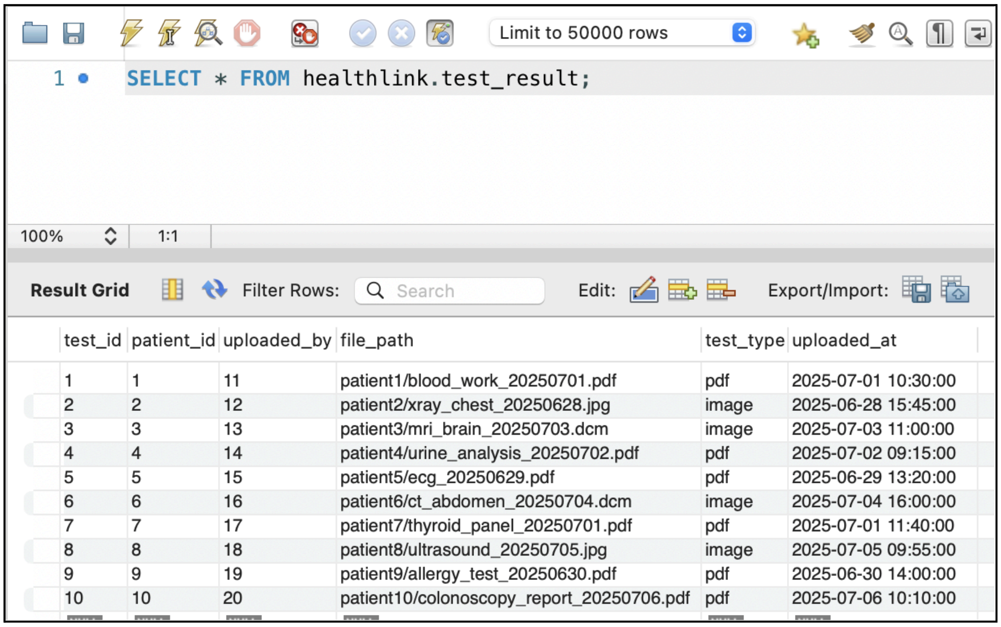
7. System_log(log_id, user_id, action, detail, timestamp)



```
SELECT * FROM healthlink.system_log;
```

log_id	user_id	action	detail	timestamp
1	5	User Login	User Tommy Liu logged in successfully.	2025-07-08 08:00:10
2	11	Medication Prescribed	Dr. Michael Chen prescribed Lisinopril to Patient…	2025-07-08 08:05:22
3	1	Profile Update	Patient Emily White updated their phone number.	2025-07-08 08:10:45
4	15	Appointment Scheduled	Dr. James Rodriguez scheduled an appointment…	2025-07-08 08:15:30
5	2	Medication Logged	Patient John Smith logged Metformin intake.	2025-07-08 08:20:15
6	18	Data Access	Dr. Olivia Martinez accessed Patient 8 records.	2025-07-08 08:25:05
7	7	Password Reset	Patient Sarah Stubbs initiated password reset.	2025-07-08 08:30:50
8	13	System Alert	Low stock alert for Atorvastatin.	2025-07-08 08:35:10
9	9	Feedback Submitted	Patient Robert Smith submitted feedback on ap…	2025-07-08 08:40:00
10	20	User Logout	Dr. Sophia Garcia logged out.	2025-07-08 08:45:20
NULL	NULL	NULL	NULL	NULL

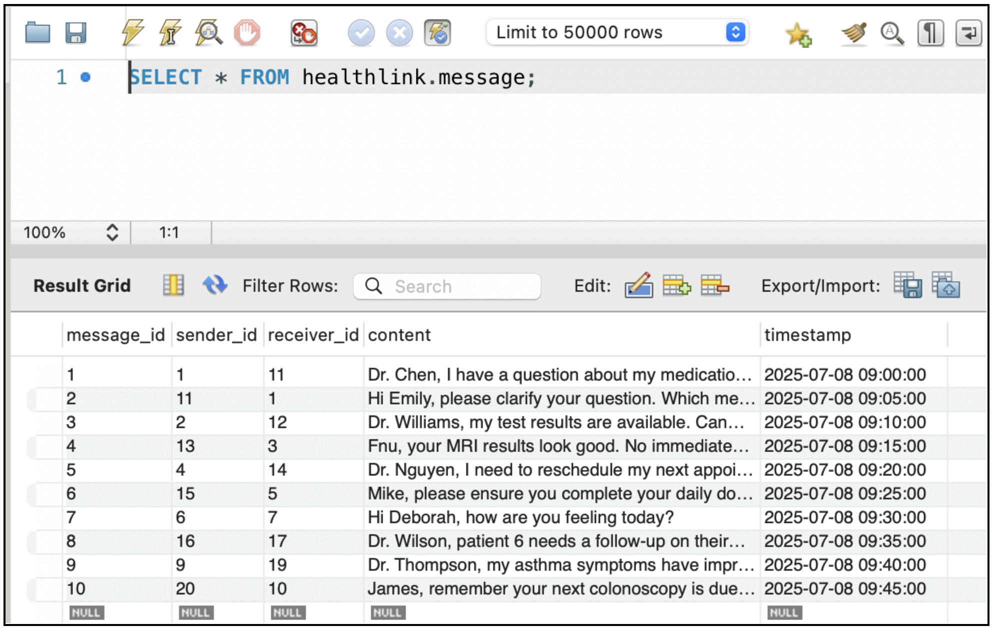
8. Test_result(test_id, patient_id, uploaded_by, file_path, test_type, uploaded_at)



```
SELECT * FROM healthlink.test_result;
```

test_id	patient_id	uploaded_by	file_path	test_type	uploaded_at
1	1	11	patient1/blood_work_20250701.pdf	pdf	2025-07-01 10:30:00
2	2	12	patient2/xray_chest_20250628.jpg	image	2025-06-28 15:45:00
3	3	13	patient3/mri_brain_20250703.dcm	image	2025-07-03 11:00:00
4	4	14	patient4/urine_analysis_20250702.pdf	pdf	2025-07-02 09:15:00
5	5	15	patient5/ecg_20250629.pdf	pdf	2025-06-29 13:20:00
6	6	16	patient6/ct_abdomen_20250704.dcm	image	2025-07-04 16:00:00
7	7	17	patient7/thyroid_panel_20250701.pdf	pdf	2025-07-01 11:40:00
8	8	18	patient8/ultrasound_20250705.jpg	image	2025-07-05 09:55:00
9	9	19	patient9/allergy_test_20250630.pdf	pdf	2025-06-30 14:00:00
10	10	20	patient10/colonoscopy_report_20250706.pdf	pdf	2025-07-06 10:10:00

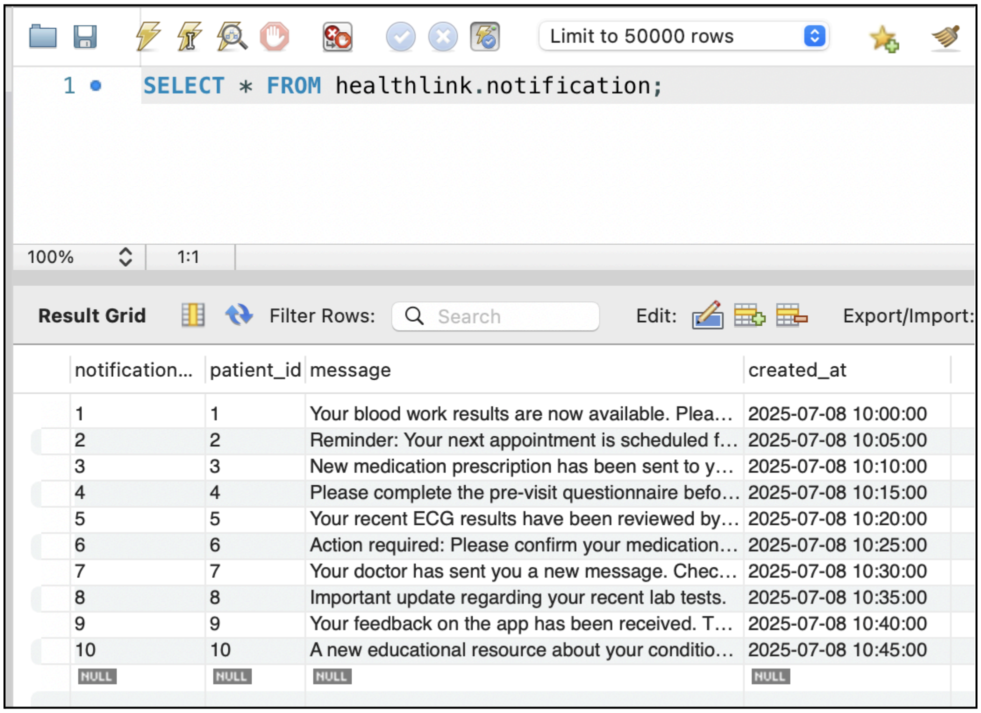
9. Message(message_id, sender_id, receiver_id, content, timestamp)



```sql
SELECT * FROM healthlink.message;
```

message_id	sender_id	receiver_id	content	timestamp
1	1	11	Dr. Chen, I have a question about my medicatio…	2025-07-08 09:00:00
2	11	1	Hi Emily, please clarify your question. Which me…	2025-07-08 09:05:00
3	2	12	Dr. Williams, my test results are available. Can…	2025-07-08 09:10:00
4	13	3	Fnu, your MRI results look good. No immediate…	2025-07-08 09:15:00
5	4	14	Dr. Nguyen, I need to reschedule my next appoi…	2025-07-08 09:20:00
6	15	5	Mike, please ensure you complete your daily do…	2025-07-08 09:25:00
7	6	7	Hi Deborah, how are you feeling today?	2025-07-08 09:30:00
8	16	17	Dr. Wilson, patient 6 needs a follow-up on their…	2025-07-08 09:35:00
9	9	19	Dr. Thompson, my asthma symptoms have impr…	2025-07-08 09:40:00
10	20	10	James, remember your next colonoscopy is due…	2025-07-08 09:45:00
NULL	NULL	NULL	NULL	NULL

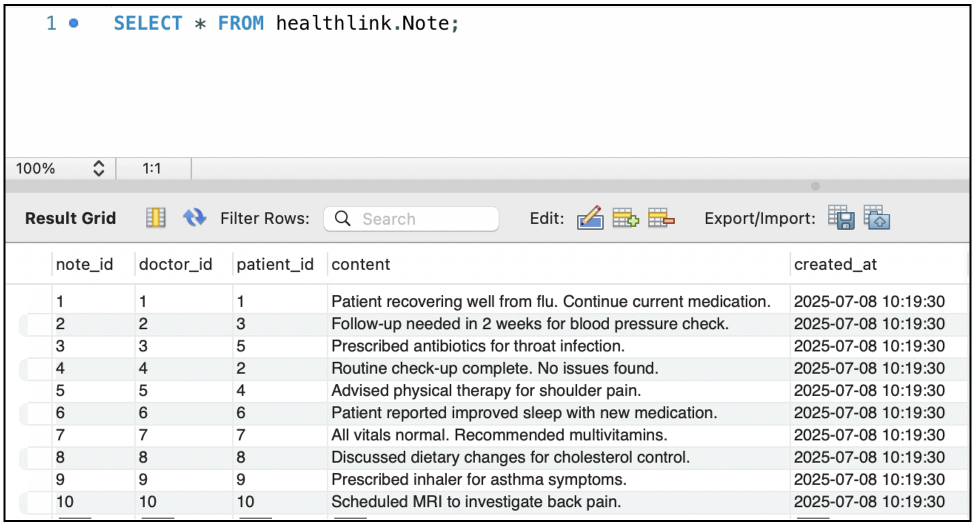
10. Notification(notification_id, patient_id, message, created_at)



```sql
SELECT * FROM healthlink.notification;
```

notification...	patient_id	message	created_at
1	1	Your blood work results are now available. Plea…	2025-07-08 10:00:00
2	2	Reminder: Your next appointment is scheduled f…	2025-07-08 10:05:00
3	3	New medication prescription has been sent to y…	2025-07-08 10:10:00
4	4	Please complete the pre-visit questionnaire befo…	2025-07-08 10:15:00
5	5	Your recent ECG results have been reviewed by…	2025-07-08 10:20:00
6	6	Action required: Please confirm your medication…	2025-07-08 10:25:00
7	7	Your doctor has sent you a new message. Chec…	2025-07-08 10:30:00
8	8	Important update regarding your recent lab tests.	2025-07-08 10:35:00
9	9	Your feedback on the app has been received. T…	2025-07-08 10:40:00
10	10	A new educational resource about your conditio…	2025-07-08 10:45:00
NULL	NULL	NULL	NULL

11. Note(note_id, doctor_id, patient_id, content, created_at)



```
SELECT * FROM healthlink.Note;
```

note_id	doctor_id	patient_id	content	created_at
1	1	1	Patient recovering well from flu. Continue current medication.	2025-07-08 10:19:30
2	2	3	Follow-up needed in 2 weeks for blood pressure check.	2025-07-08 10:19:30
3	3	5	Prescribed antibiotics for throat infection.	2025-07-08 10:19:30
4	4	2	Routine check-up complete. No issues found.	2025-07-08 10:19:30
5	5	4	Advised physical therapy for shoulder pain.	2025-07-08 10:19:30
6	6	6	Patient reported improved sleep with new medication.	2025-07-08 10:19:30
7	7	7	All vitals normal. Recommended multivitamins.	2025-07-08 10:19:30
8	8	8	Discussed dietary changes for cholesterol control.	2025-07-08 10:19:30
9	9	9	Prescribed inhaler for asthma symptoms.	2025-07-08 10:19:30
10	10	10	Scheduled MRI to investigate back pain.	2025-07-08 10:19:30