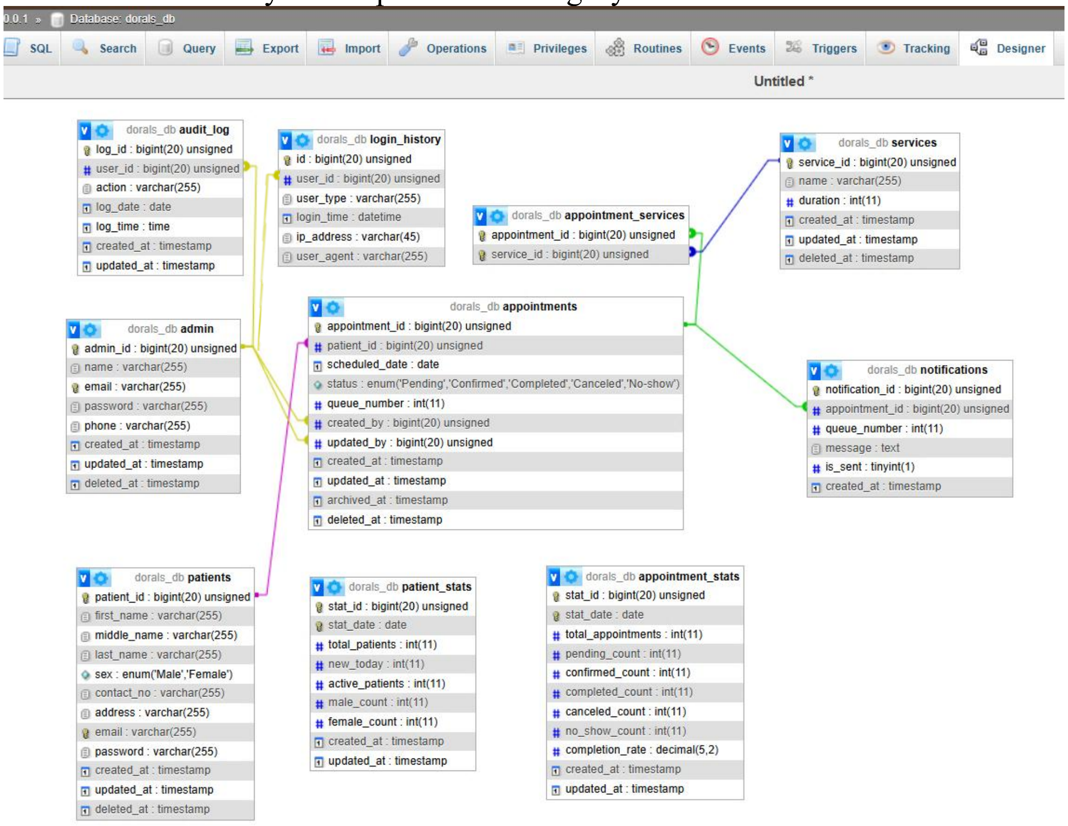
Isabela State University
College of Computing Studies, Information and
Communication Technology Information Technology

Kathryn Jasmine S. Ingiaen
Nicole Faye Carabbacan
BSIT 2-1 WMAD

These are required features that must be implemented on your system/database

- **Normalization**

The database schema for the D-ORALS system has been designed with normalization in mind to eliminate redundancy and improve data integrity.



dorals_db **audit_log**
log_id : bigint(20) unsigned
user_id : bigint(20) unsigned
action : varchar(255)
log_date : date
log_time : time
created_at : timestamp
updated_at : timestamp

dorals_db **login_history**
id : bigint(20) unsigned
user_id : bigint(20) unsigned
user_type : varchar(255)
login_time : datetime
ip_address : varchar(45)
user_agent : varchar(255)

dorals_db **appointment_services**
appointment_id : bigint(20) unsigned
service_id : bigint(20) unsigned

dorals_db **services**
service_id : bigint(20) unsigned
name : varchar(255)
duration : int(11)
created_at : timestamp
updated_at : timestamp
deleted_at : timestamp

dorals_db **admin**
admin_id : bigint(20) unsigned
name : varchar(255)
email : varchar(255)
password : varchar(255)
phone : varchar(255)
created_at : timestamp
updated_at : timestamp
deleted_at : timestamp

dorals_db **appointments**
appointment_id : bigint(20) unsigned
patient_id : bigint(20) unsigned
scheduled_date : date
status : enum('Pending','Confirmed','Completed','Canceled','No-show')
queue_number : int(11)
created_by : bigint(20) unsigned
updated_by : bigint(20) unsigned
created_at : timestamp
updated_at : timestamp
archived_at : timestamp
deleted_at : timestamp

dorals_db **notifications**
notification_id : bigint(20) unsigned
appointment_id : bigint(20) unsigned
queue_number : int(11)
message : text
is_sent : tinyint(1)
created_at : timestamp

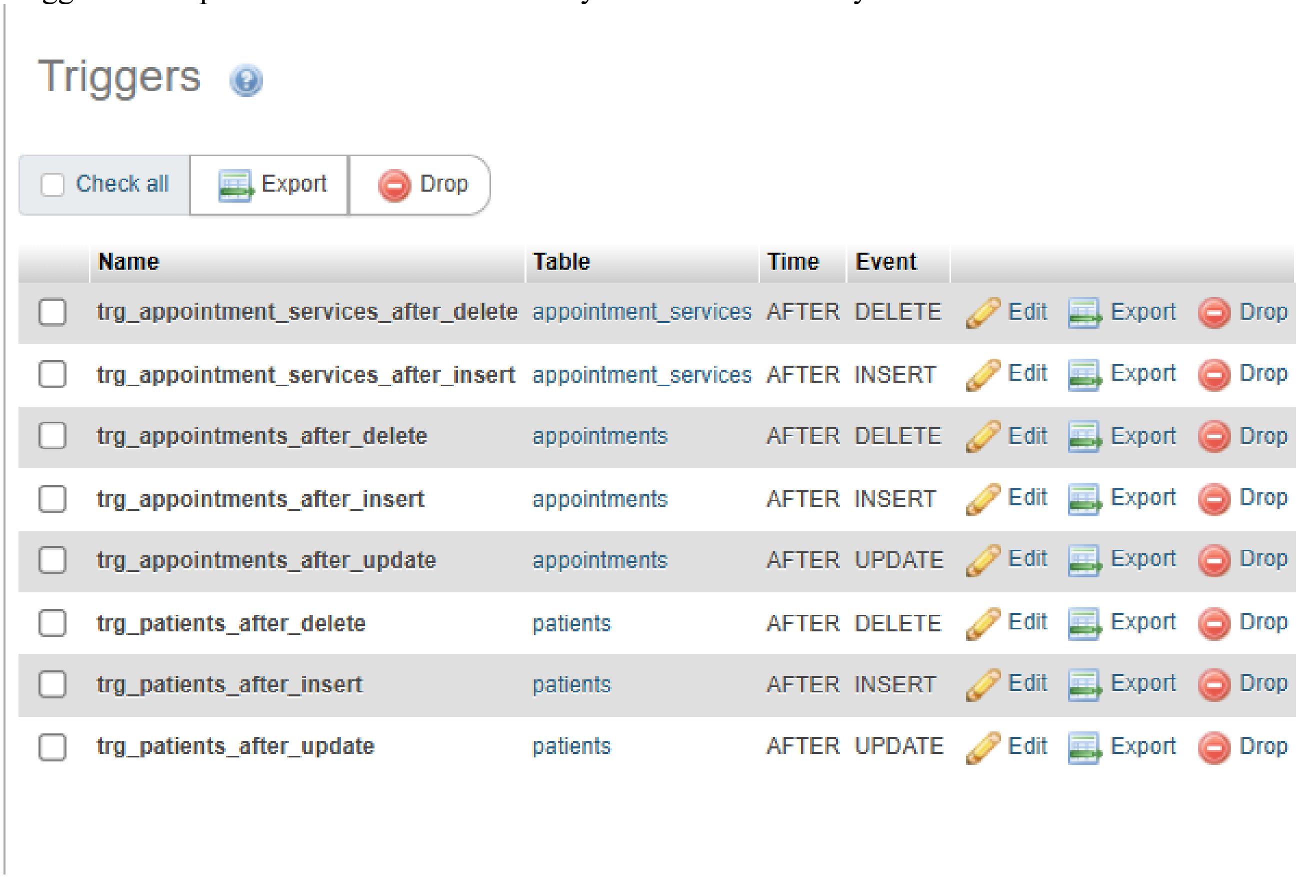
dorals_db **patients**
patient_id : bigint(20) unsigned
first_name : varchar(255)
middle_name : varchar(255)
last_name : varchar(255)
sex : enum('Male','Female')
contact_no : varchar(255)
address : varchar(255)
email : varchar(255)
password : varchar(255)
created_at : timestamp
updated_at : timestamp
deleted_at : timestamp

dorals_db **patient_stats**
stat_id : bigint(20) unsigned
stat_date : date
total_patients : int(11)
new_today : int(11)
active_patients : int(11)
male_count : int(11)
female_count : int(11)
created_at : timestamp
updated_at : timestamp

dorals_db **appointment_stats**
stat_id : bigint(20) unsigned
stat_date : date
total_appointments : int(11)
pending_count : int(11)
confirmed_count : int(11)
completed_count : int(11)
canceled_count : int(11)
no_show_count : int(11)
completion_rate : decimal(5,2)
created_at : timestamp
updated_at : timestamp

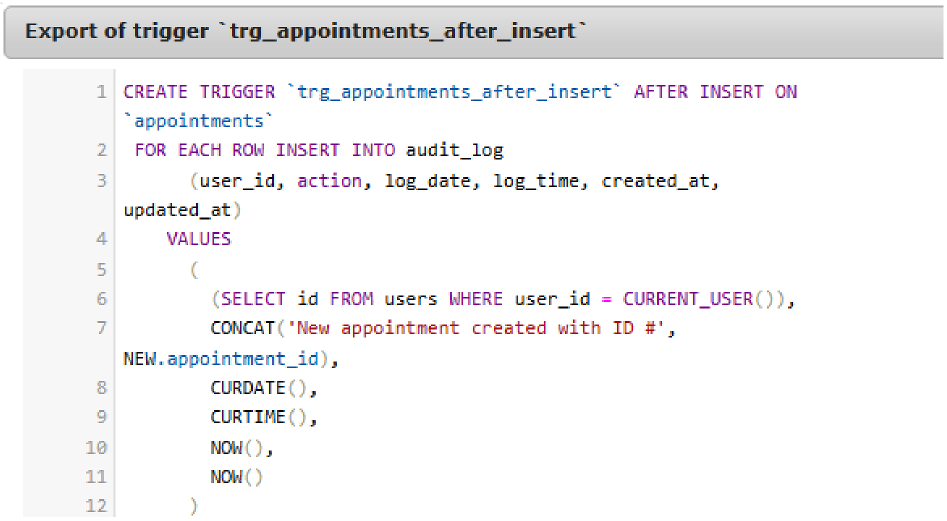
- **Trigger**

Triggers are implemented in the D-ORALS system to automate key actions in the database.



	Name	Table	Time	Event			
□	trg_appointment_services_after_delete	appointment_services	AFTER	DELETE	Edit	Export	Drop
□	trg_appointment_services_after_insert	appointment_services	AFTER	INSERT	Edit	Export	Drop
□	trg_appointments_after_delete	appointments	AFTER	DELETE	Edit	Export	Drop
□	trg_appointments_after_insert	appointments	AFTER	INSERT	Edit	Export	Drop
□	trg_appointments_after_update	appointments	AFTER	UPDATE	Edit	Export	Drop
□	trg_patients_after_delete	patients	AFTER	DELETE	Edit	Export	Drop
□	trg_patients_after_insert	patients	AFTER	INSERT	Edit	Export	Drop
□	trg_patients_after_update	patients	AFTER	UPDATE	Edit	Export	Drop

For example, a trigger is used to automatically update the patient appointment status once a scheduled appointment has passed, ensuring that records are always up-to-date and consistent without manual intervention

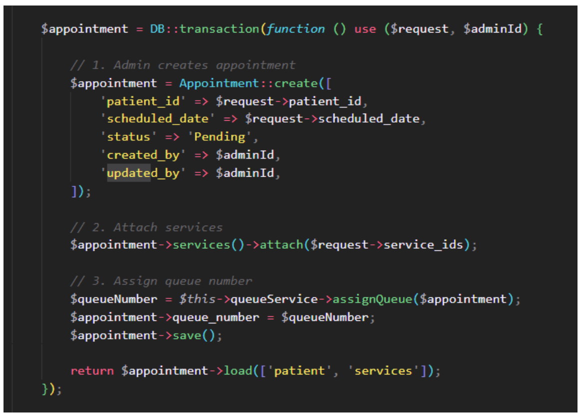


Export of trigger `trg_appointments_after_insert`

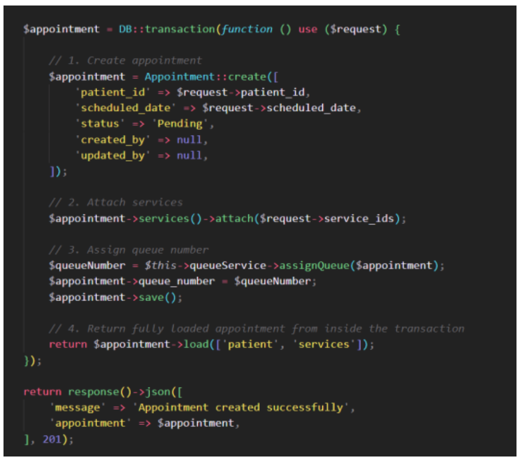
```sql
CREATE TRIGGER `trg_appointments_after_insert` AFTER INSERT ON `appointments`
 FOR EACH ROW INSERT INTO audit_log
     (user_id, action, log_date, log_time, created_at, updated_at)
    VALUES
     (
       (SELECT id FROM users WHERE user_id = CURRENT_USER()),
       CONCAT('New appointment created with ID #', NEW.appointment_id),
       CURDATE(),
       CURTIME(),
       NOW(),
       NOW()
     )
```

- **Transaction**

The D-ORALS system uses database transactions to ensure that patient registration and appointment scheduling are executed atomically.



```php
$appointment = DB::transaction(function () use ($request, $adminId) {

    // 1. Admin creates appointment
    $appointment = Appointment::create([
        'patient_id' => $request->patient_id,
        'scheduled_date' => $request->scheduled_date,
        'status' => 'Pending',
        'created_by' => $adminId,
        'updated_by' => $adminId,
    ]);

    // 2. Attach services
    $appointment->services()->attach($request->service_ids);

    // 3. Assign queue number
    $queueNumber = $this->queueService->assignQueue($appointment);
    $appointment->queue_number = $queueNumber;
    $appointment->save();

    return $appointment->load(['patient', 'services']);
});
```

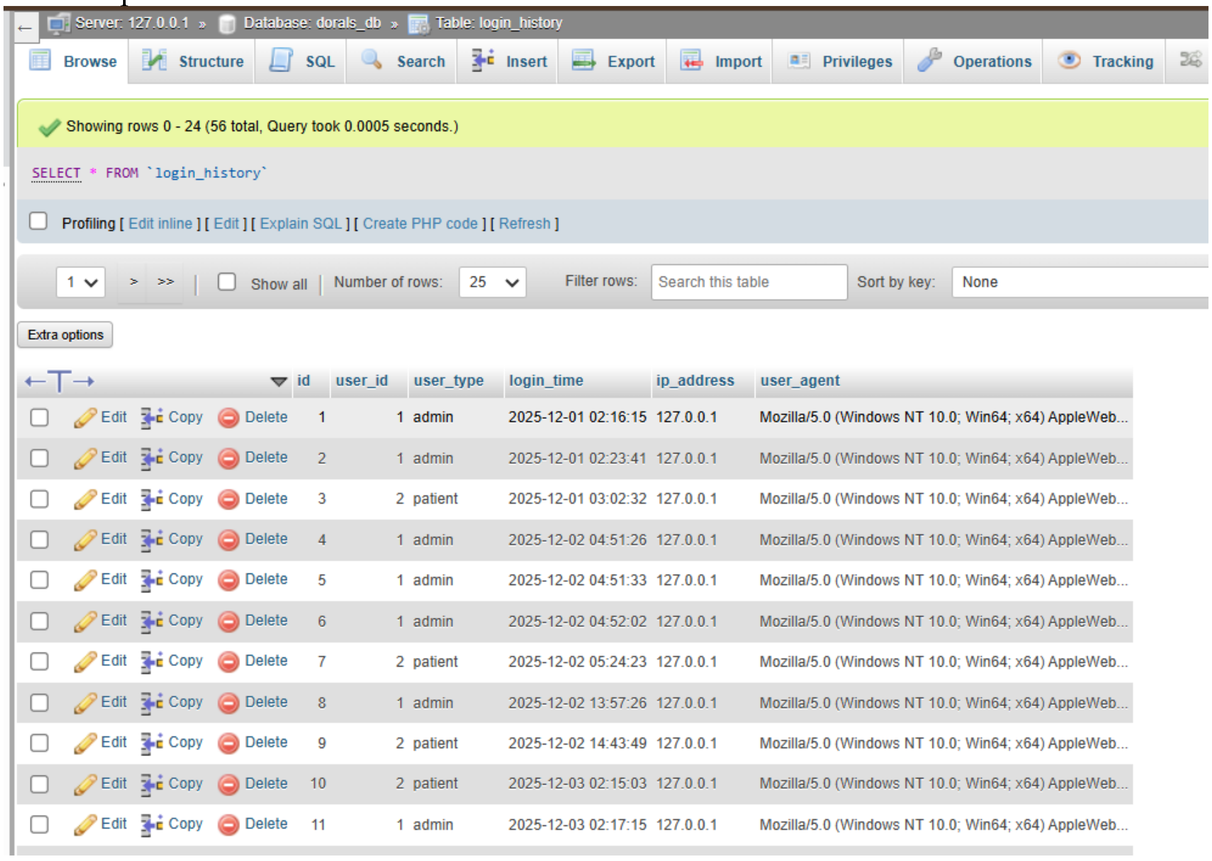


```php
$appointment = DB::transaction(function () use ($request) {

    // 1. Create appointment
    $appointment = Appointment::create([
        'patient_id' => $request->patient_id,
        'scheduled_date' => $request->scheduled_date,
        'status' => 'Pending',
        'created_by' => null,
        'updated_by' => null,
    ]);

    // 2. Attach services
    $appointment->services()->attach($request->service_ids);

    // 3. Assign queue number
    $queueNumber = $this->queueService->assignQueue($appointment);
    $appointment->queue_number = $queueNumber;
    $appointment->save();

    // 4. Return fully loaded appointment from inside the transaction
    return $appointment->load(['patient', 'services']);
});

return response()->json([
    'message' => 'Appointment created successfully',
    'appointment' => $appointment,
], 201);
```

The transaction in creating *appointment*, is attached to adding *services* and assigning *queue number,* ensure that if any part of the process fails, the entire operation is rolled back, maintaining the consistency of the database and ensuring both the patient and appointment tables are updated together.

- **Login History**

The system tracks and logs all user login attempts, including successful and failed logins, along with timestamps



Server: 127.0.0.1 » Database: dorals_db » Table: login_history
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking

✓ Showing rows 0 - 24 (56 total, Query took 0.0005 seconds.)

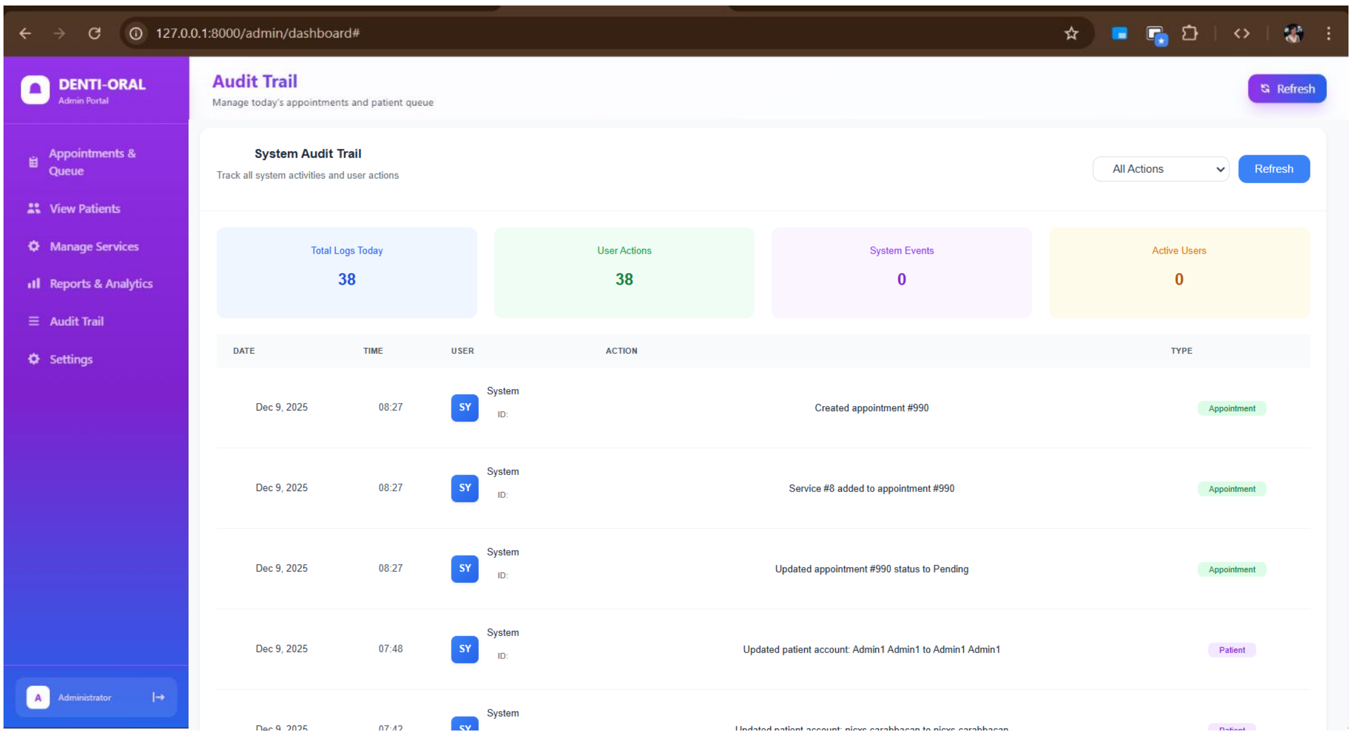
```sql
SELECT * FROM `login_history`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
1 > >> □ Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None
Extra options

←T→				id	user_id	user_type	login_time	ip_address	user_agent
□	Edit	Copy	Delete	1	1	admin	2025-12-01 02:16:15	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb...
□	Edit	Copy	Delete	2	1	admin	2025-12-01 02:23:41	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb...
□	Edit	Copy	Delete	3	2	patient	2025-12-01 03:02:32	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb...
□	Edit	Copy	Delete	4	1	admin	2025-12-02 04:51:26	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb...
□	Edit	Copy	Delete	5	1	admin	2025-12-02 04:51:33	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb...
□	Edit	Copy	Delete	6	1	admin	2025-12-02 04:52:02	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb...
□	Edit	Copy	Delete	7	2	patient	2025-12-02 05:24:23	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb...
□	Edit	Copy	Delete	8	1	admin	2025-12-02 13:57:26	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb...
□	Edit	Copy	Delete	9	2	patient	2025-12-02 14:43:49	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb...
□	Edit	Copy	Delete	10	2	patient	2025-12-03 02:15:03	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb...
□	Edit	Copy	Delete	11	1	admin	2025-12-03 02:17:15	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb...

- Audit Logs

The D-ORALS system keeps detailed audit logs of all user actions, including changes to patient records, appointment scheduling, and any administrative modifications. These logs are connected to the triggers



Audit Trail

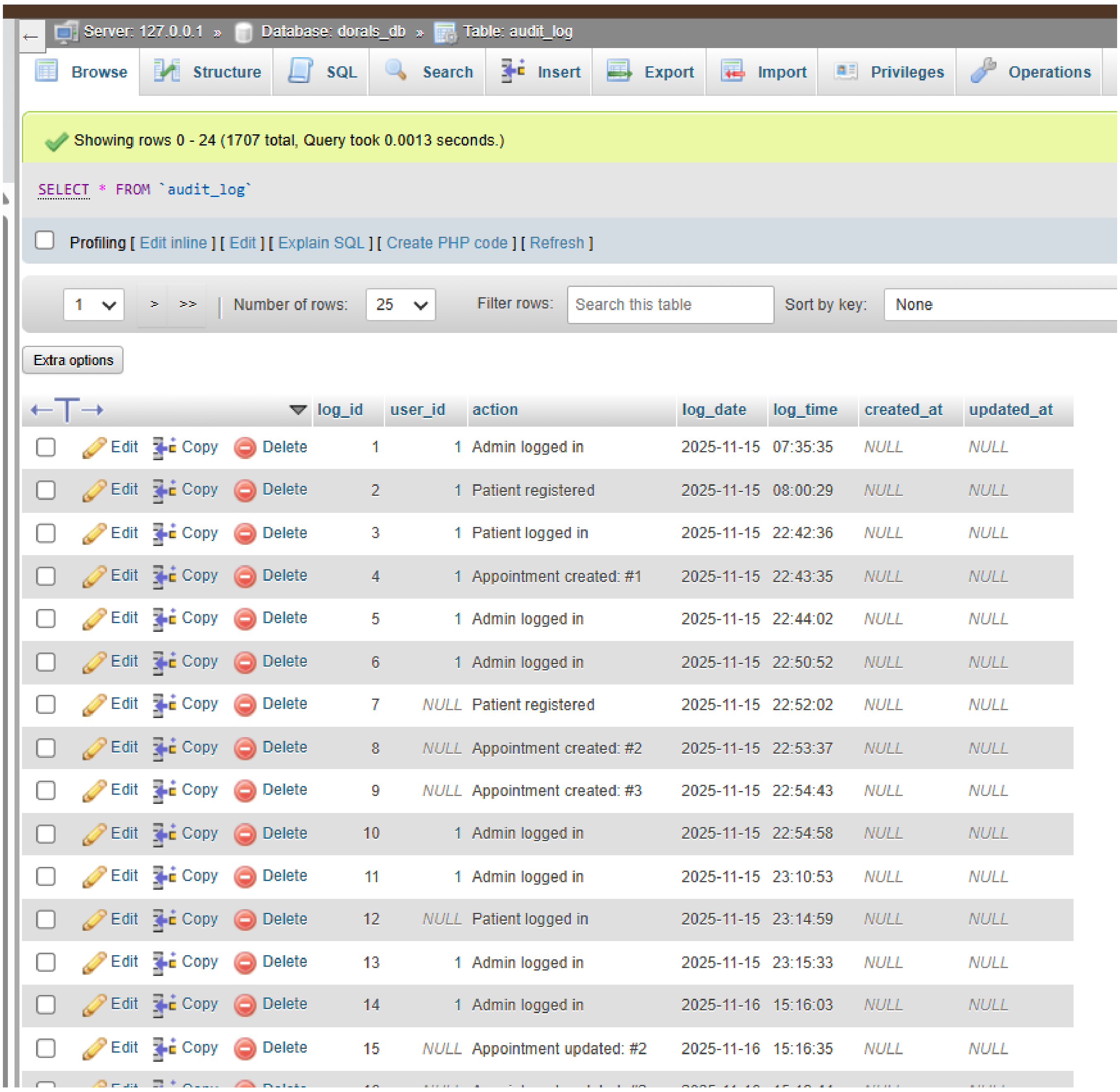
Manage today's appointments and patient queue

System Audit Trail

Track all system activities and user actions

Total Logs Today	User Actions	System Events	Active Users
38	38	0	0

DATE	TIME	USER	ACTION	TYPE
Dec 9, 2025	08:27	SY System ID:	Created appointment #990	Appointment
Dec 9, 2025	08:27	SY System ID:	Service #8 added to appointment #990	Appointment
Dec 9, 2025	08:27	SY System ID:	Updated appointment #990 status to Pending	Appointment
Dec 9, 2025	07:48	SY System ID:	Updated patient account: Admin1 Admin1 to Admin1 Admin1	Patient
Dec 9, 2025	07:42	SY System	Updated patient account: nicxs carabbacan to nicxs carabbacan	Patient



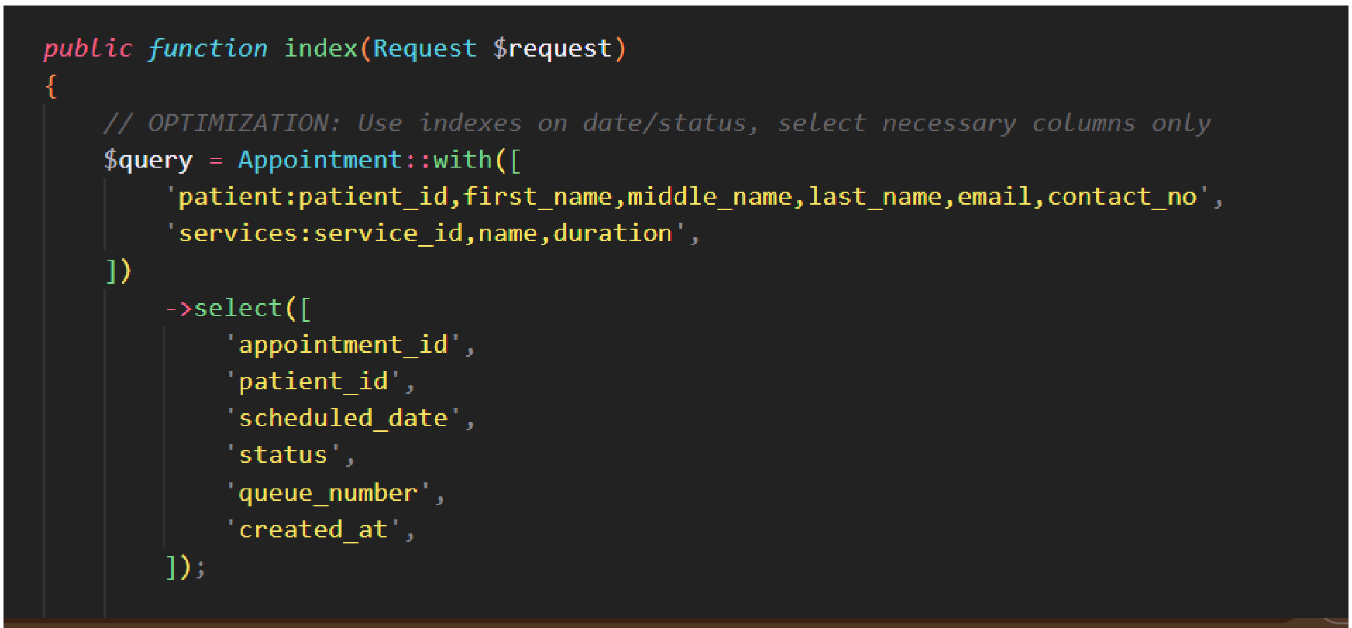
Server: 127.0.0.1 » Database: dorals_db » Table: audit_log
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
1
> >>
Number of rows: 25
Filter rows: Search this table
Sort by key: None
Extra options

Showing rows 0 - 24 (1707 total, Query took 0.0013 seconds.)

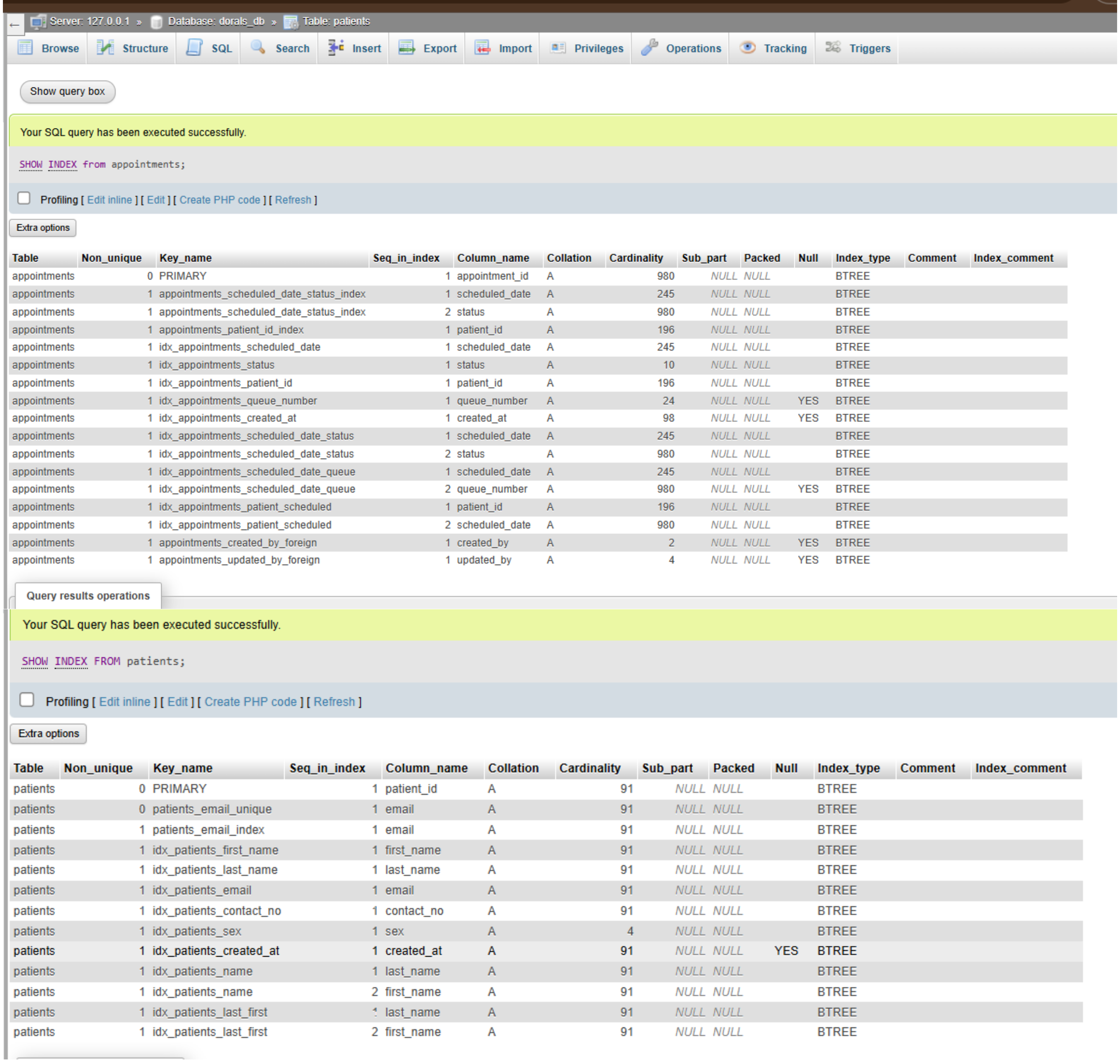
```
SELECT * FROM `audit_log`
```

log_id	user_id	action	log_date	log_time	created_at	updated_at
1	1	Admin logged in	2025-11-15	07:35:35	NULL	NULL
2	1	Patient registered	2025-11-15	08:00:29	NULL	NULL
3	1	Patient logged in	2025-11-15	22:42:36	NULL	NULL
4	1	Appointment created: #1	2025-11-15	22:43:35	NULL	NULL
5	1	Admin logged in	2025-11-15	22:44:02	NULL	NULL
6	1	Admin logged in	2025-11-15	22:50:52	NULL	NULL
7	NULL	Patient registered	2025-11-15	22:52:02	NULL	NULL
8	NULL	Appointment created: #2	2025-11-15	22:53:37	NULL	NULL
9	NULL	Appointment created: #3	2025-11-15	22:54:43	NULL	NULL
10	1	Admin logged in	2025-11-15	22:54:58	NULL	NULL
11	1	Admin logged in	2025-11-15	23:10:53	NULL	NULL
12	NULL	Patient logged in	2025-11-15	23:14:59	NULL	NULL
13	1	Admin logged in	2025-11-15	23:15:33	NULL	NULL
14	1	Admin logged in	2025-11-16	15:16:03	NULL	NULL
15	NULL	Appointment updated: #2	2025-11-16	15:16:35	NULL	NULL

- **Optimization**
  - **index**



```
public function index(Request $request)
{
    // OPTIMIZATION: Use indexes on date/status, select necessary columns only
    $query = Appointment::with([
        'patient:patient_id,first_name,middle_name,last_name,email,contact_no',
        'services:service_id,name,duration',
    ])
        ->select([
            'appointment_id',
            'patient_id',
            'scheduled_date',
            'status',
            'queue_number',
            'created_at',
        ]);
```



Server: 127.0.0.1 » Database: dorals_db » Table: patients
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
Show query box

Your SQL query has been executed successfully.

```
SHOW INDEX from appointments;
```

Profiling [Edit inline] [Edit] [Create PHP code] [Refresh]
Extra options

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
appointments	0	PRIMARY	1	appointment_id	A	980	NULL	NULL		BTREE		
appointments	1	appointments_scheduled_date_status_index	1	scheduled_date	A	245	NULL	NULL		BTREE		
appointments	1	appointments_scheduled_date_status_index	2	status	A	980	NULL	NULL		BTREE		
appointments	1	appointments_patient_id_index	1	patient_id	A	196	NULL	NULL		BTREE		
appointments	1	idx_appointments_scheduled_date	1	scheduled_date	A	245	NULL	NULL		BTREE		
appointments	1	idx_appointments_status	1	status	A	10	NULL	NULL		BTREE		
appointments	1	idx_appointments_patient_id	1	patient_id	A	196	NULL	NULL		BTREE		
appointments	1	idx_appointments_queue_number	1	queue_number	A	24	NULL	NULL	YES	BTREE		
appointments	1	idx_appointments_created_at	1	created_at	A	98	NULL	NULL	YES	BTREE		
appointments	1	idx_appointments_scheduled_date_status	1	scheduled_date	A	245	NULL	NULL		BTREE		
appointments	1	idx_appointments_scheduled_date_status	2	status	A	980	NULL	NULL		BTREE		
appointments	1	idx_appointments_scheduled_date_queue	1	scheduled_date	A	245	NULL	NULL		BTREE		
appointments	1	idx_appointments_scheduled_date_queue	2	queue_number	A	980	NULL	NULL	YES	BTREE		
appointments	1	idx_appointments_patient_scheduled	1	patient_id	A	196	NULL	NULL		BTREE		
appointments	1	idx_appointments_patient_scheduled	2	scheduled_date	A	980	NULL	NULL		BTREE		
appointments	1	appointments_created_by_foreign	1	created_by	A	2	NULL	NULL	YES	BTREE		
appointments	1	appointments_updated_by_foreign	1	updated_by	A	4	NULL	NULL	YES	BTREE		

Your SQL query has been executed successfully.

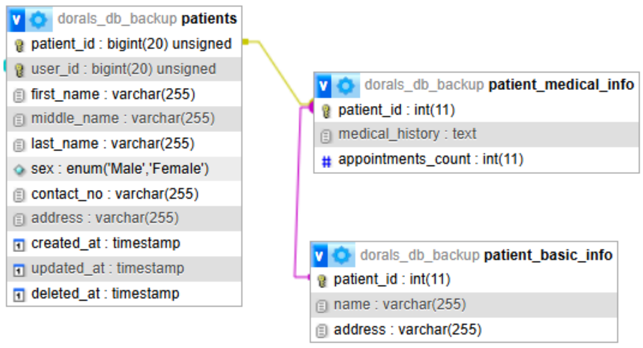
```
SHOW INDEX FROM patients;
```

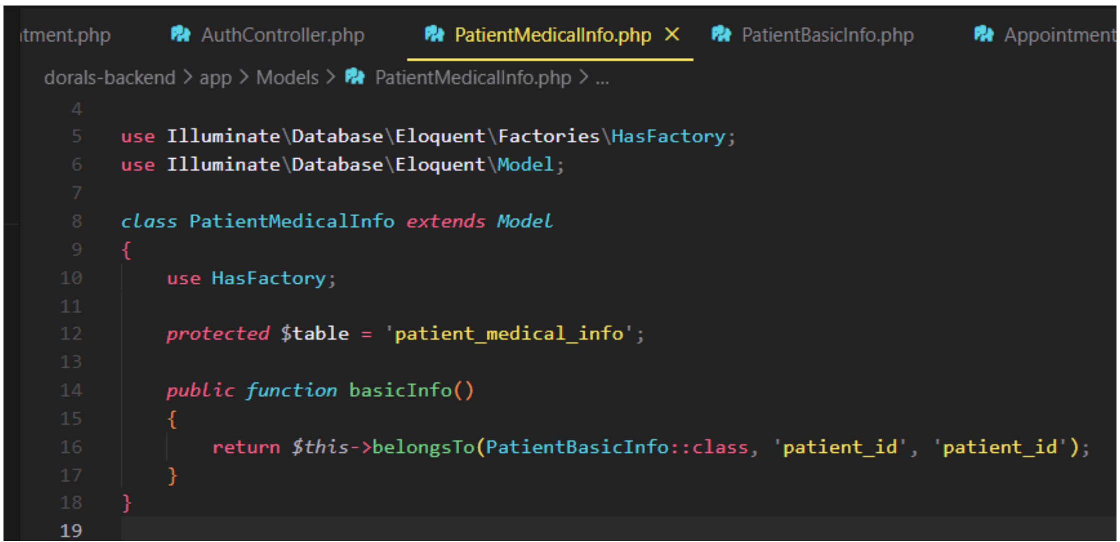
Profiling [Edit inline] [Edit] [Create PHP code] [Refresh]
Extra options

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
patients	0	PRIMARY	1	patient_id	A	91	NULL	NULL		BTREE		
patients	0	patients_email_unique	1	email	A	91	NULL	NULL		BTREE		
patients	1	patients_email_index	1	email	A	91	NULL	NULL		BTREE		
patients	1	idx_patients_first_name	1	first_name	A	91	NULL	NULL		BTREE		
patients	1	idx_patients_last_name	1	last_name	A	91	NULL	NULL		BTREE		
patients	1	idx_patients_email	1	email	A	91	NULL	NULL		BTREE		
patients	1	idx_patients_contact_no	1	contact_no	A	91	NULL	NULL		BTREE		
patients	1	idx_patients_sex	1	sex	A	4	NULL	NULL		BTREE		
patients	1	idx_patients_created_at	1	created_at	A	91	NULL	NULL	YES	BTREE		
patients	1	idx_patients_name	1	last_name	A	91	NULL	NULL		BTREE		
patients	1	idx_patients_name	2	first_name	A	91	NULL	NULL		BTREE		
patients	1	idx_patients_last_first	1	last_name	A	91	NULL	NULL		BTREE		
patients	1	idx_patients_last_first	2	first_name	A	91	NULL	NULL		BTREE		

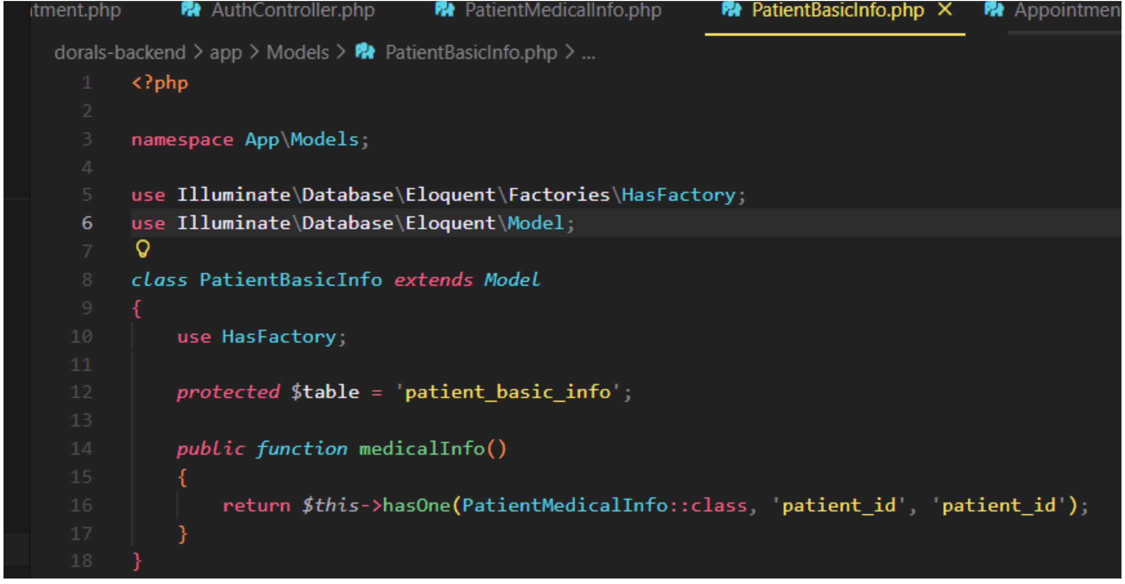
  - **vertical and horizontal partition**

the patients table contains many columns (e.g., patient_id, name, address, phone_number, medical_history, appointments_count, etc.). If only a few columns are frequently queried (e.g., name and address), you can create two partitions for the table.


dorals_db_backup patients
patient_id : bigint(20) unsigned
user_id : bigint(20) unsigned
first_name : varchar(255)
middle_name : varchar(255)
last_name : varchar(255)
sex : enum('Male','Female')
contact_no : varchar(255)
address : varchar(255)
created_at : timestamp
updated_at : timestamp
deleted_at : timestamp
dorals_db_backup patient_medical_info
patient_id : int(11)
medical_history : text
appointments_count : int(11)
dorals_db_backup patient_basic_info
patient_id : int(11)
name : varchar(255)
address : varchar(255)



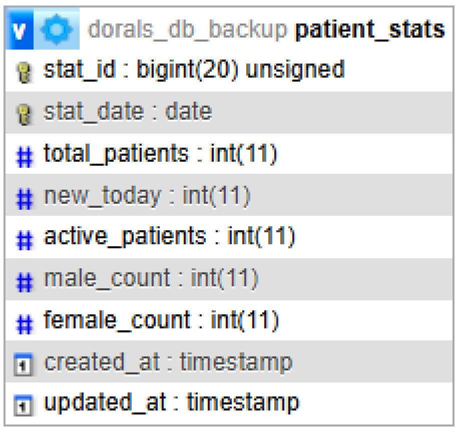
```php

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class PatientMedicalInfo extends Model
{
    use HasFactory;

    protected $table = 'patient_medical_info';

    public function basicInfo()
    {
        return $this->belongsTo(PatientBasicInfo::class, 'patient_id', 'patient_id');
    }
}

```



```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class PatientBasicInfo extends Model
{
    use HasFactory;

    protected $table = 'patient_basic_info';

    public function medicalInfo()
    {
        return $this->hasOne(PatientMedicalInfo::class, 'patient_id', 'patient_id');
    }
}
```

- **Aggregate**

find the total number of appointments for each patient, total of female and male count also.

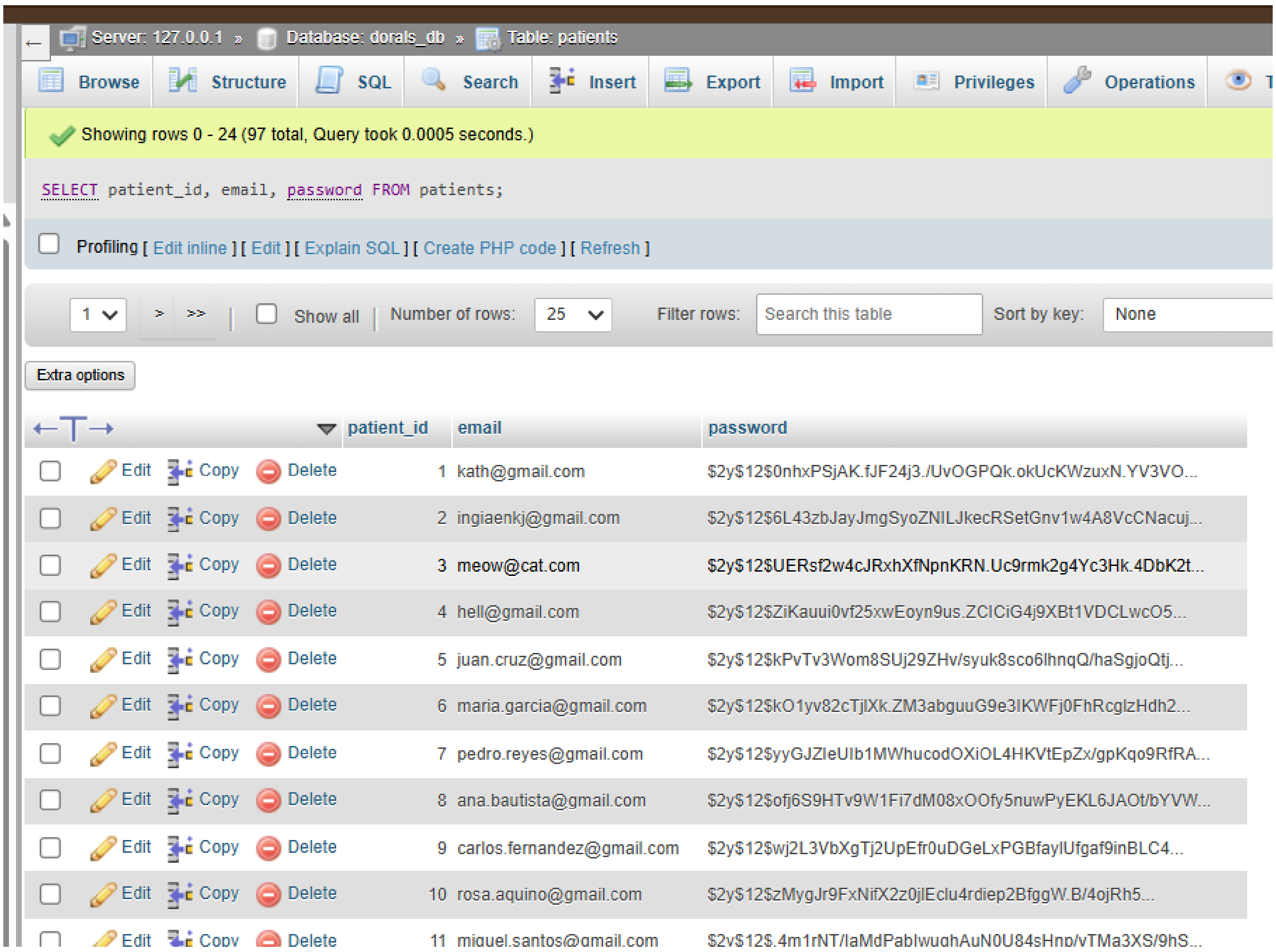


dorals_db_backup **patient_stats**

stat_id : bigint(20) unsigned
stat_date : date
total_patients : int(11)
new_today : int(11)
active_patients : int(11)
male_count : int(11)
female_count : int(11)
created_at : timestamp
updated_at : timestamp

- **Encryption**

Sensitive data, such as patient personal information and passwords, are encrypted both at rest and in transit. The D-ORALS system uses industry-standard encryption techniques to ensure that personal data remains secure and inaccessible to unauthorized users


Server: 127.0.0.1 » Database: dorals_db » Table: patients
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
T

Showing rows 0 - 24 (97 total, Query took 0.0005 seconds.)

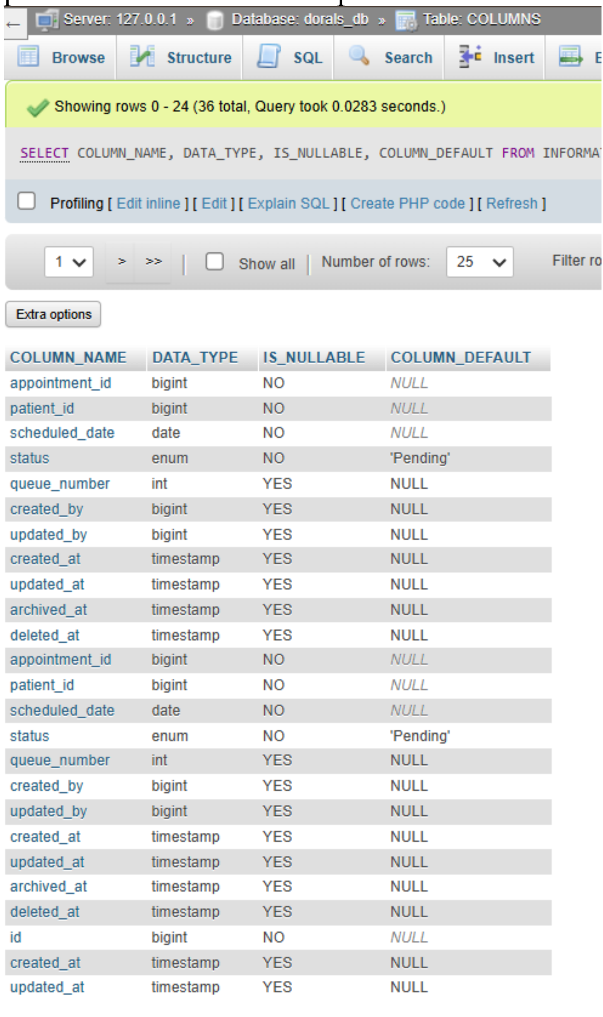
```
SELECT patient_id, email, password FROM patients;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
1
>
>>
Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None
Extra options

	patient_id	email	password
□ Edit Copy Delete	1	kath@gmail.com	$2y$12$0nhxPSjAK.fJF24j3./UvOGPQk.okUcKWzuxN.YV3VO...
□ Edit Copy Delete	2	ingiaenkj@gmail.com	$2y$12$6L43zbJayJmgSyoZNILJkecRSetGnv1w4A8VcCNacuj...
□ Edit Copy Delete	3	meow@cat.com	$2y$12$UERsf2w4cJRxhXfNpnKRN.Uc9rmk2g4Yc3Hk.4DbK2t...
□ Edit Copy Delete	4	hell@gmail.com	$2y$12$ZiKauui0vf25xwEoyn9us.ZClCiG4j9XBt1VDCLwcO5...
□ Edit Copy Delete	5	juan.cruz@gmail.com	$2y$12$kPvTv3Wom8SUj29ZHv/syuk8sco6lhnqQ/haSgjoQtj...
□ Edit Copy Delete	6	maria.garcia@gmail.com	$2y$12$kO1yv82cTjlXk.ZM3abguuG9e3IKWFj0FhRcglzHdh2...
□ Edit Copy Delete	7	pedro.reyes@gmail.com	$2y$12$yyGJZleUlb1MWhucodOXiOL4HKVtEpZx/gpKqo9RfRA...
□ Edit Copy Delete	8	ana.bautista@gmail.com	$2y$12$ofj6S9HTv9W1Fi7dM08xOOfy5nuwPyEKL6JAOt/bYVW...
□ Edit Copy Delete	9	carlos.fernandez@gmail.com	$2y$12$wj2L3VbXgTj2UpEfr0uDGeLxPGBfaylUfgaf9inBLC4...
□ Edit Copy Delete	10	rosa.aquino@gmail.com	$2y$12$zMygJr9FxNifX2z0jlEclu4rdiep2BfggW.B/4ojRh5...
□ Edit Copy Delete	11	miguel.santos@gmail.com	$2y$12$.4m1rNT/laMdPablwughAuN0U84sHnp/vTMa3XS/9hS...

- **Domain Constraints**

The system implements domain constraints on key data fields to ensure that data entered into the system is valid. For example, appointment dates must fall within the clinic's operating hours, and patient IDs must follow a predefined format to maintain consistency and avoid errors.

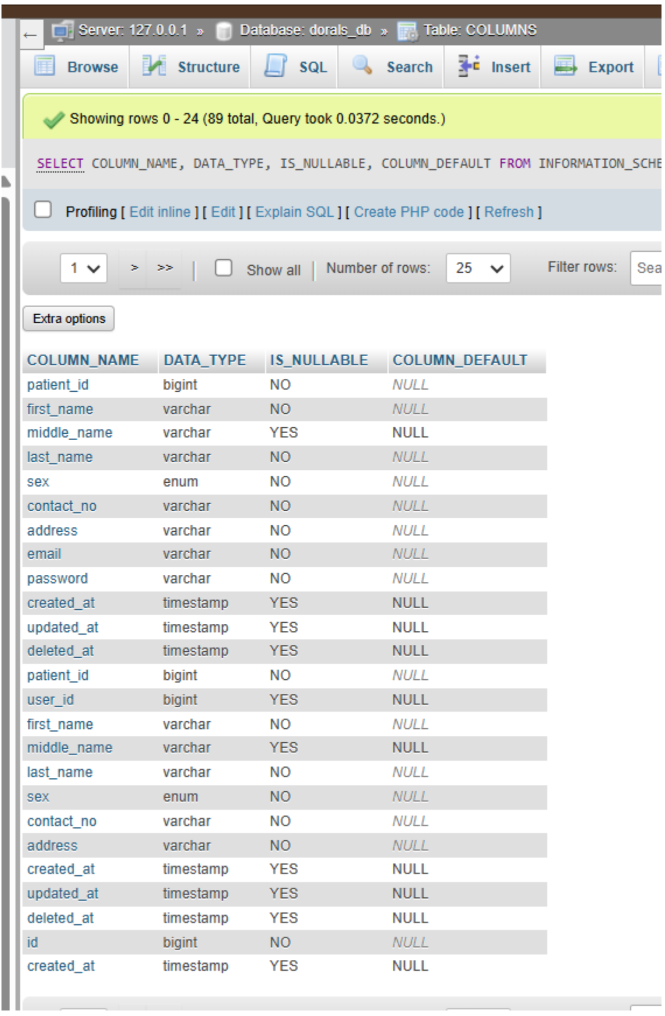

Server: 127.0.0.1 » Database: dorals_db » Table: COLUMNS
Browse
Structure
SQL
Search
Insert
E

Showing rows 0 - 24 (36 total, Query took 0.0283 seconds.)

```
SELECT COLUMN_NAME, DATA_TYPE, IS_NULLABLE, COLUMN_DEFAULT FROM INFORMA
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
1
>
>>
Show all
Number of rows: 25
Filter ro
Extra options

COLUMN_NAME	DATA_TYPE	IS_NULLABLE	COLUMN_DEFAULT
appointment_id	bigint	NO	NULL
patient_id	bigint	NO	NULL
scheduled_date	date	NO	NULL
status	enum	NO	'Pending'
queue_number	int	YES	NULL
created_by	bigint	YES	NULL
updated_by	bigint	YES	NULL
created_at	timestamp	YES	NULL
updated_at	timestamp	YES	NULL
archived_at	timestamp	YES	NULL
deleted_at	timestamp	YES	NULL
appointment_id	bigint	NO	NULL
patient_id	bigint	NO	NULL
scheduled_date	date	NO	NULL
status	enum	NO	'Pending'
queue_number	int	YES	NULL
created_by	bigint	YES	NULL
updated_by	bigint	YES	NULL
created_at	timestamp	YES	NULL
updated_at	timestamp	YES	NULL
archived_at	timestamp	YES	NULL
deleted_at	timestamp	YES	NULL
id	bigint	NO	NULL
created_at	timestamp	YES	NULL
updated_at	timestamp	YES	NULL

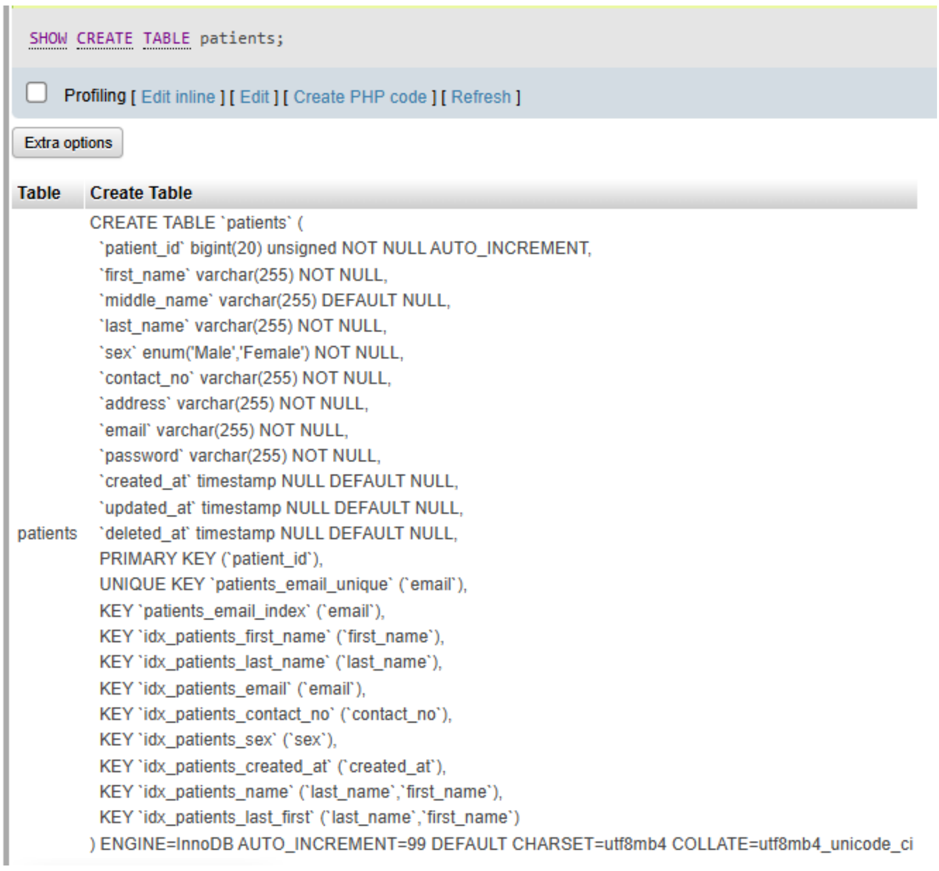

Server: 127.0.0.1 » Database: dorals_db » Table: COLUMNS
Browse
Structure
SQL
Search
Insert
Export

Showing rows 0 - 24 (89 total, Query took 0.0372 seconds.)

```
SELECT COLUMN_NAME, DATA_TYPE, IS_NULLABLE, COLUMN_DEFAULT FROM INFORMATION_SCHE
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
1
>
>>
Show all
Number of rows: 25
Filter rows: Sea
Extra options

COLUMN_NAME	DATA_TYPE	IS_NULLABLE	COLUMN_DEFAULT
patient_id	bigint	NO	NULL
first_name	varchar	NO	NULL
middle_name	varchar	YES	NULL
last_name	varchar	NO	NULL
sex	enum	NO	NULL
contact_no	varchar	NO	NULL
address	varchar	NO	NULL
email	varchar	NO	NULL
password	varchar	NO	NULL
created_at	timestamp	YES	NULL
updated_at	timestamp	YES	NULL
deleted_at	timestamp	YES	NULL
patient_id	bigint	NO	NULL
user_id	bigint	YES	NULL
first_name	varchar	NO	NULL
middle_name	varchar	YES	NULL
last_name	varchar	NO	NULL
sex	enum	NO	NULL
contact_no	varchar	NO	NULL
address	varchar	NO	NULL
created_at	timestamp	YES	NULL
updated_at	timestamp	YES	NULL
deleted_at	timestamp	YES	NULL
id	bigint	NO	NULL
created_at	timestamp	YES	NULL

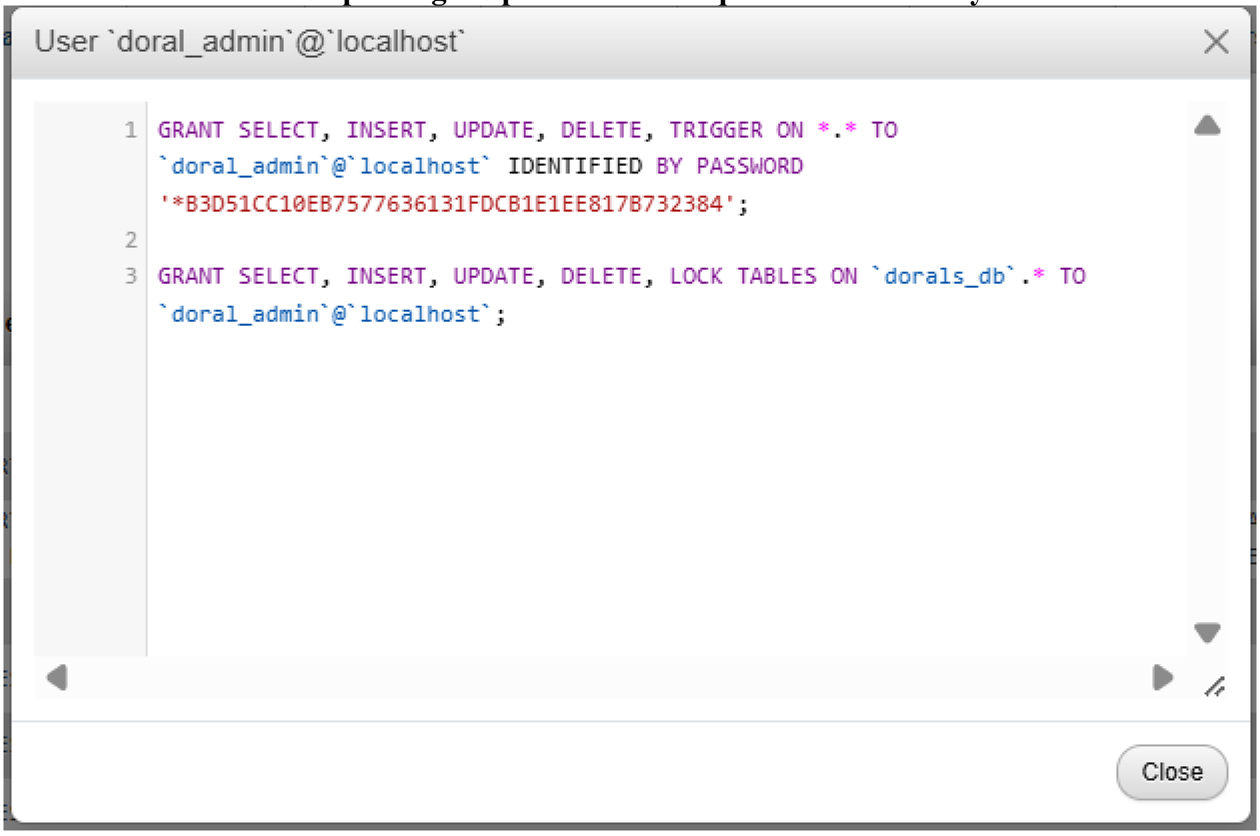


```
SHOW CREATE TABLE patients;
```

□ Profiling [Edit inline] [Edit] [Create PHP code] [Refresh]

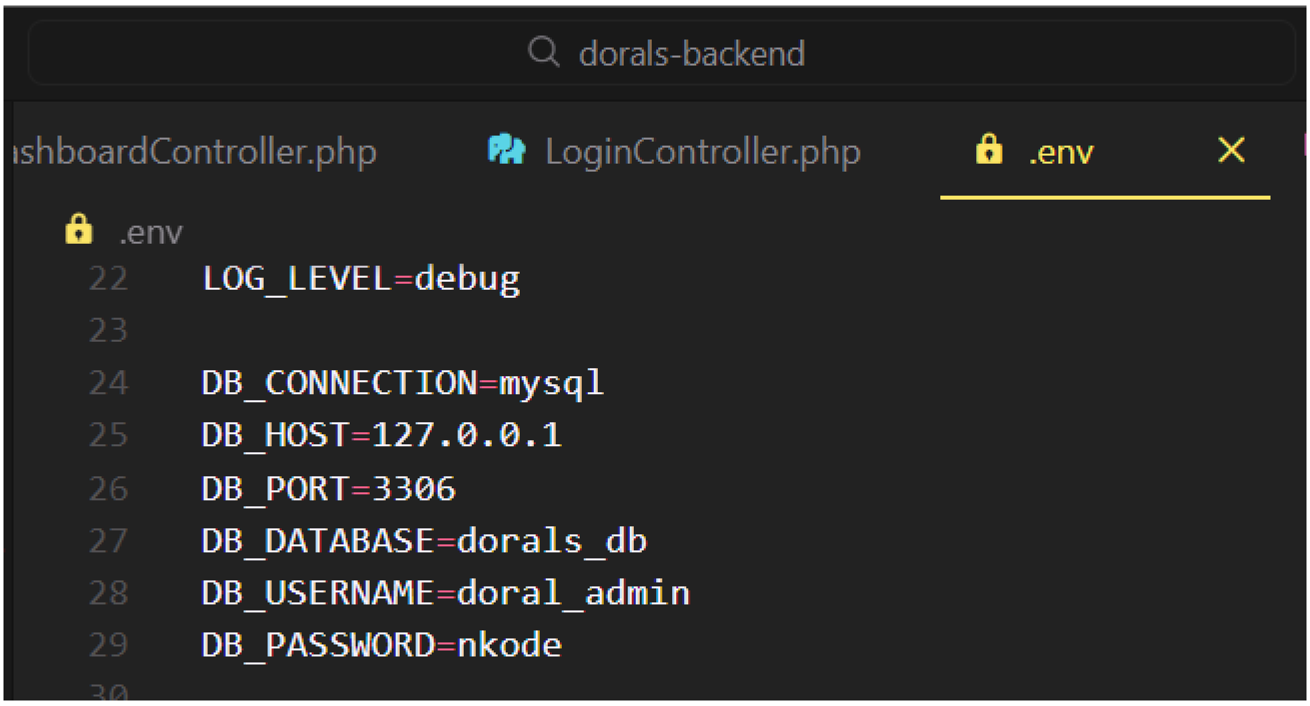
Table	Create Table
patients	CREATE TABLE \`patients\` (\`patient_id\` bigint(20) unsigned NOT NULL AUTO_INCREMENT, \`first_name\` varchar(255) NOT NULL, \`middle_name\` varchar(255) DEFAULT NULL, \`last_name\` varchar(255) NOT NULL, \`sex\` enum('Male','Female') NOT NULL, \`contact_no\` varchar(255) NOT NULL, \`address\` varchar(255) NOT NULL, \`email\` varchar(255) NOT NULL, \`password\` varchar(255) NOT NULL, \`created_at\` timestamp NULL DEFAULT NULL, \`updated_at\` timestamp NULL DEFAULT NULL, \`deleted_at\` timestamp NULL DEFAULT NULL, PRIMARY KEY (\`patient_id\`), UNIQUE KEY \`patients_email_unique\` (\`email\`), KEY \`patients_email_index\` (\`email\`), KEY \`idx_patients_first_name\` (\`first_name\`), KEY \`idx_patients_last_name\` (\`last_name\`), KEY \`idx_patients_email\` (\`email\`), KEY \`idx_patients_contact_no\` (\`contact_no\`), KEY \`idx_patients_sex\` (\`sex\`), KEY \`idx_patients_created_at\` (\`created_at\`), KEY \`idx_patients_name\` (\`last_name\`,\`first_name\`), KEY \`idx_patients_last_first\` (\`last_name\`,\`first_name\`)) ENGINE=InnoDB AUTO_INCREMENT=99 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci

- **Limit Database User privilege depends on the requirements of the system**



User \`doral_admin\`@\`localhost\`

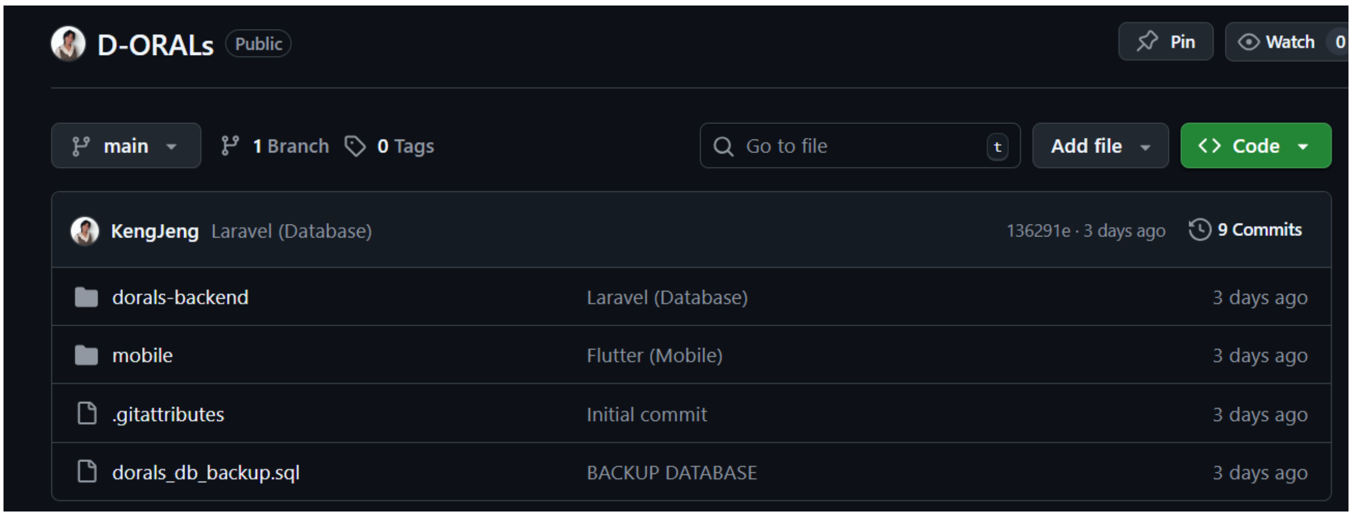
```
GRANT SELECT, INSERT, UPDATE, DELETE, TRIGGER ON *.* TO
`doral_admin`@`localhost` IDENTIFIED BY PASSWORD
'*B3D51CC10EB7577636131FDCB1E1EE817B732384';

GRANT SELECT, INSERT, UPDATE, DELETE, LOCK TABLES ON `dorals_db`.* TO
`doral_admin`@`localhost`;
```



```
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=dorals_db
DB_USERNAME=doral_admin
DB_PASSWORD=nkode
```

- **Recovery and Backup (demonstration, plan or integrate to the system)**

We export our database from the MySQL and the system project files of Laravel, Vue, and Flutter folders, and uploaded it at the Github.



KengJeng Laravel (Database) 136291e · 3 days ago 9 Commits

dorals-backend	Laravel (Database)	3 days ago
mobile	Flutter (Mobile)	3 days ago
.gitattributes	Initial commit	3 days ago
dorals_db_backup.sql	BACKUP DATABASE	3 days ago