

# Sri Lanka Institute of Information Technology



SE2032 – Database Management Systems

Project Report



Group – 25: EduGlobe Website Database System

StudentID	Name
IT24101493	Yasiru Silva
IT24101499	H.R.M.D.C.B Herath
IT24101515	M.G.N. D Kumara
IT24101636	M.D.C Yavindi

BSc (Hons) in Computer Science Y2 S1

October 2025

## Table of Contents

List of Figures .....	3
List of Tables .....	4
1. Introduction .....	5
1.1 Client Details: .....	5
1.2 Project Overview .....	5
1.3 Members' Scope .....	5
2. Database Design Requirements .....	6
3. ER/EER MODEL .....	7
3.1 ER/EER Model Diagram .....	7
3.2 Mapping the ER/EER Model to Relational Schema .....	8
3.3 Data Dictionary .....	9
4. Database Implementation .....	17
4.1 Database Creation .....	17
4.2 Populating Data .....	21
4.3 Display Tables .....	25
4.4 SQL QUERIES .....	29
5 Conclusion .....	37

## List of Figures

Figure 1: ER/EER diagram .....	7
Figure 2 : relational mapping diagram .....	8

## List of Tables

Table 1: Student Table-Data Dictionary .....	9
Table 2: Admin Table-Data Dictionary .....	10
Table 3:Universities Table-Data Dictionary.....	11
Table 4:Programs Table-Data Dictionary.....	12
Table 5:Application Table-Data Dictionary .....	14
Table 6 : Contact Message Table-Data Dictionary.....	14
Table 7 Payment Table Table-Data Dictionary .....	15
Table 8 Notification Log Table-Data Dictionary .....	16

# 1. Introduction

## Topic: SIT International Student Management System

### 1.1 Client Details:

- **Client Name:** SIT International (Pvt) Ltd
- **Owners:** Mr. Subash Mannapperuma,  
Mr. Imanka Wimalasiri  
Mr. Thanuja Mallikarachchi
- **Business Type:** Educational Consultancy
- **Address:** No:135/3/20 EL Senanayake St, Kandy
- **Contact Email:** [sit.intsl@gmail.com](mailto:sit.intsl@gmail.com)
- **Contact Phone:** 0702635042

### 1.2 Project Overview

SIT International Student Management System is a comprehensive web-based system designed to automate and streamline student recruitment and application to SIT International. The system tracks the whole student life cycle from preliminary inquiry and program selection to application, payment, and communication. The digitization eliminates paperwork, reduces errors, and provides smoother experience for students and SIT staff.

### 1.3 Members' Scope

Each member handles both front-end and back-end development for their assigned module:

- **IT24101499:** User and Admin Management (Student/Admin registration, login, profiles)
- **IT24101515:** Program Management (University and program catalog)
- **IT24101636:** Application and Contact Management
- **IT24101493:** Payment and Notification Management

## **2. Database Design Requirements**

### **For User & Admin Management (IT24101499):**

- Students can be registered with their personal details (name, email, phone, address, etc.)
- Both admins and students can log in through email and password
- Admins can view, activate, deactivate, and update all user accounts
- System distinguishes based on various user roles (Student, Admin with subtypes)

### **For Program Management (IT24101515):**

- Admins want to Add, View, Edit, and Delete universities and programs
- Programs hold data like university name, program name, degree type
- Students are able to browse and search for available programs and universities
- Programs are linked to specific universities

### **For Application & Contact Management (IT24101636):**

- Students are able to apply to programs by submitting online forms along with documents
- Applications have traceable status (SUBMITTED, UNDER\_REVIEW, APPROVED)
- Admins want to view applications and assign handlers
- Students are able to send queries, and admins can respond via the system.

### **For Payment & Notification Management (IT24101493):**

- Students are able to pay fees on their application
- Payment history records quantity, type, status, and transaction data
- Admins are able to view and modify payment status
- System sends payments and application status updates and notifications

### 3. ER/EER MODEL

#### 3.1 ER/EER Model Diagram

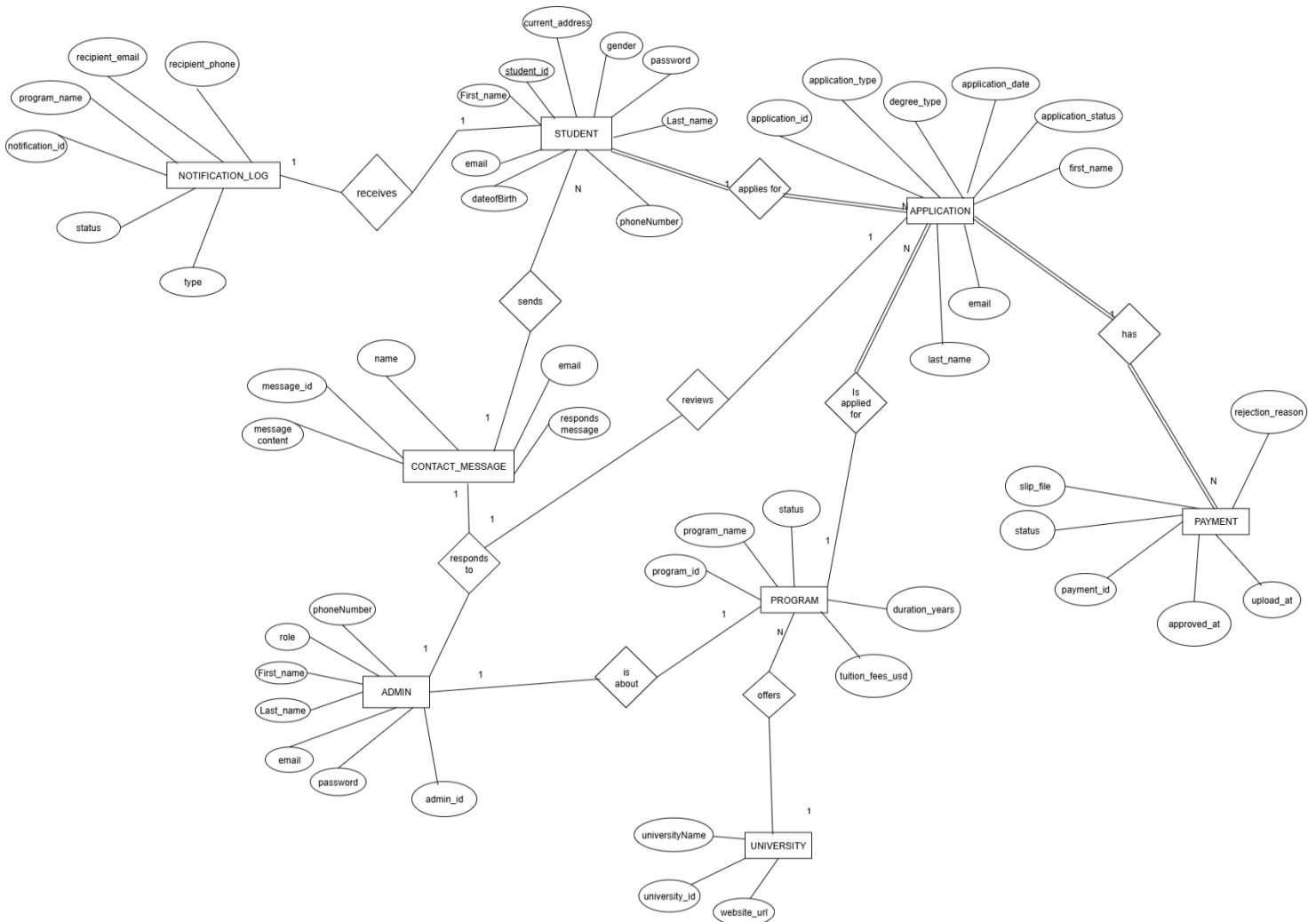


Figure 1: ER/EER diagram

## 3.2 Mapping the ER/EER Model to Relational Schema

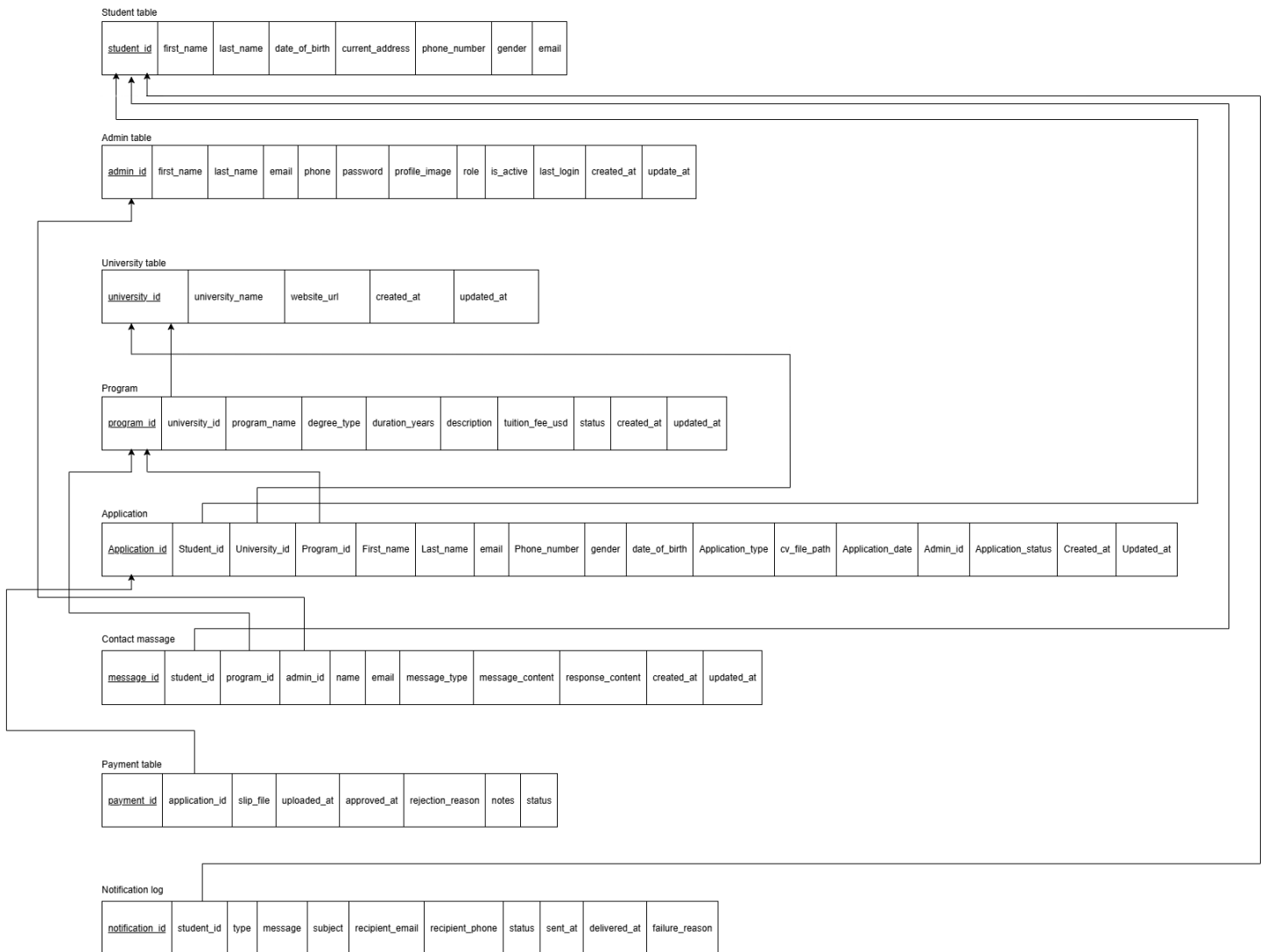


Figure 2 : relational mapping diagram



### 3.3 Data Dictionary

#### User and Admin Management (IT24101499)

##### Student Table

Field Name	Data Type	Constraints	Description
student_id	INT	PK	Unique ID for each student.
first_name	VARCHAR(100)	NOT NULL	Student's first name.
last_name	VARCHAR(100)	NOT NULL	Student's last name.
date_of_birth	DATE	NULL	Student's date of birth.
current_address	VARCHAR(255)	NULL	Student's current address.
phone_number	VARCHAR(20)	NULL	Student's contact number.
gender	VARCHAR(10)	NULL	Student's gender.
email	VARCHAR(150)	NOT NULL, UNIQUE	Student's email address (unique).

*Table 1: Student Table-Data Dictionary*

## Admin Table

Field Name	Data Type	Constraints	Description
admin_id	INT	PK	Unique ID for each admin.
first_name	VARCHAR(100)	NOT NULL	Admin's first name.
last_name	VARCHAR(100)	NULL	Admin's last name.
email	VARCHAR(100)	NOT NULL, UNIQUE	Admin's email address.
phone	VARCHAR(20)	NULL	Admin's contact phone number.
password	VARCHAR(255)	NOT NULL	Admin's account password (hashed).
profile_image	VARCHAR(500)	NULL	Path to profile image.
role	ENUM('ADMIN', 'MANAGER', 'STAFF')	NOT NULL, DEFAULT 'STAFF'	Role of the admin in the system.
is_active	BOOLEAN	NOT NULL, DEFAULT TRUE	Indicates if admin account is active.
last_login	DATETIME	NULL	Timestamp of last login.
created_at	DATETIME	NOT NULL	Record creation timestamp.
updated_at	DATETIME	NOT NULL	Record update timestamp.

Table 2: Admin Table-Data Dictionary

## Program Management (IT24101515)

### Universities Table

Field Name	Data Type	Constraints	Description
university_id	INT	PK	Unique ID for each university.
university_name	VARCHAR(200)	NOT NULL	Name of the university.
website_url	VARCHAR(255)	NULL	Official website of the university.
created_at	DATETIME	DEFAULT CURRENT_TIMESTAMP	Record creation timestamp.
updated_at	DATETIME	DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	Record last update timestamp.

Table 3: Universities Table-Data Dictionary

## Program Table

Field Name	Data Type	Constraints	Description
program_id	INT	PK	Unique ID for each program.
university_id	INT	FK → universities(university_id)	University offering the program.
program_name	VARCHAR(255)	NOT NULL	Name of the academic program.
degree_type	ENUM('BACHELOR', 'MASTER', 'PHD')	NOT NULL	Type of degree offered.
duration_years	INT	NOT NULL	Duration of the program in years.
description	TEXT	NULL	Detailed description of the program.
tuition_fee_usd	DECIMAL(10,2)	NULL	Tuition fee in USD.
status	ENUM('ACTIVE', 'INACTIVE')	NOT NULL, DEFAULT 'ACTIVE'	Program's active status.
created_at	DATETIME	DEFAULT CURRENT_TIMESTAMP	Record creation timestamp.
updated_at	DATETIME	DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	Record last update timestamp.

Table 4: Programs Table-Data Dictionary

## Application and Contact Management (IT24101636)

**Applications Table**

Field Name	Data Type	Constraints	Description
application_id	INT	PK	Unique ID for each application.
student_id	INT	FK → students(student_id)	Student submitting the application.
university_id	INT	FK → universities(university_id)	University applied to.
program_id	INT	FK → programs(program_id)	Program applied for.
first_name	VARCHAR(100)	NOT NULL	Applicant's first name.
last_name	VARCHAR(100)	NOT NULL	Applicant's last name.
email	VARCHAR(150)	NOT NULL	Applicant's email address.
phone_number	VARCHAR(20)	NULL	Applicant's contact number.
gender	VARCHAR(10)	NULL	Applicant's gender.
date_of_birth	DATE	NULL	Applicant's date of birth.
application_type	ENUM('BACHELOR', 'SPECIALIST', 'MASTER', 'PHD')	NOT NULL	Type of application.
cv_file_path	VARCHAR(500)	NULL	File path of uploaded CV.
application_date	DATETIME	NOT NULL	Date of application submission.
admin_id	INT	FK → admins(admin_id)	Admin who processed the application.
application_status	ENUM('PENDING', 'APPROVED', 'REJECTED')	NOT NULL, DEFAULT 'PENDING'	Application's current status.
created_at	DATETIME	DEFAULT CURRENT_TIMESTAMP	Record creation timestamp.

updated_at	DATETIME	DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	Record last update timestamp.
------------	----------	--	-------------------------------

Table 5: Application Table-Data Dictionary

### contact messages

Field Name	Data Type	Constraints	Description
message_id	INT	PK	Unique ID for each message.
student_id	INT	FK → students(student_id)	Student sending the message (if applicable).
program_id	INT	FK → programs(program_id)	Related program (if applicable).
admin_id	INT	FK → admins(admin_id)	Admin responding to message (if applicable).
name	VARCHAR(100)	NOT NULL	Name of the sender.
email	VARCHAR(150)	NOT NULL	Sender's email address.
message_type	ENUM('PROGRAM_INFO', 'SCHOLARSHIP_INFO', 'OTHER')	NOT NULL	Type/category of the message.
message_content	TEXT	NOT NULL	Message content.
response_content	TEXT	NULL	Admin's response message.
created_at	DATETIME	DEFAULT CURRENT_TIMESTAMP	Record creation timestamp.
updated_at	DATETIME	DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	Record last update timestamp.

Table 6 : Contact Message Table-Data Dictionary

## Payment and Notification Management (IT24101493)

### Payments Table

Field Name	Data Type	Constraints	Description
payment_id	INT	PK	Unique ID for each payment record.
application_id	INT	FK → applications(application_id)	Related application ID.
slip_file	VARCHAR(500)	NOT NULL	Path to uploaded payment slip.
uploaded_at	DATETIME	DEFAULT CURRENT_TIMESTAMP	Timestamp when payment was uploaded.
approved_at	DATETIME	NULL	Timestamp when payment was approved.
rejection_reason	VARCHAR(500)	NULL	Reason for rejection (if any).
notes	VARCHAR(500)	NULL	Additional admin notes.
status	ENUM('PENDING', 'APPROVED', 'REJECTED')	NOT NULL, DEFAULT 'PENDING'	Payment verification status.

Table 7 Payment Table Table-Data Dictionary

## Notification logs

Field Name	Data Type	Constraints	Description
notification_id	INT	PK	Unique ID for each notification.
student_id	INT	FK → students(student_id)	Student who received the notification.
type	ENUM('APPLICATION_UPDATE', 'PAYMENT_CONFIRMATION', 'GENERAL')	NOT NULL	Type of notification.
message	TEXT	NOT NULL	Notification message content.
subject	VARCHAR(255)	NULL	Notification subject.
recipient_email	VARCHAR(150)	NULL	Recipient's email address.
recipient_phone	VARCHAR(20)	NULL	Recipient's phone number.
status	ENUM('PENDING', 'SENT', 'DELIVERED', 'FAILED')	NOT NULL, DEFAULT 'PENDING'	Delivery status of the notification.
sent_at	DATETIME	DEFAULT CURRENT_TIMESTAMP	Time notification was sent.
delivered_at	DATETIME	NULL	Time notification was delivered.
failure_reason	TEXT	NULL	Reason for failed delivery (if any).

Table 8 Notification Log Table-Data Dictionary

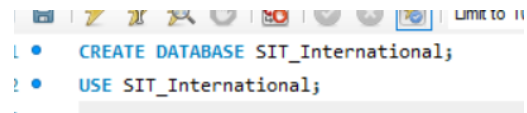


## 4. Database Implementation

### 4.1 Database Creation

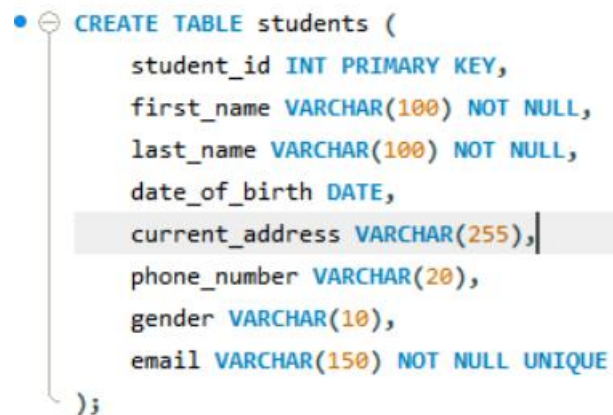
#### User and Admin Management (IT24101493)

Creating database



```
1 • CREATE DATABASE SIT_International;  
2 • USE SIT_International;  
.
```

Students Table



```
• CREATE TABLE students (  
    student_id INT PRIMARY KEY,  
    first_name VARCHAR(100) NOT NULL,  
    last_name VARCHAR(100) NOT NULL,  
    date_of_birth DATE,  
    current_address VARCHAR(255),  
    phone_number VARCHAR(20),  
    gender VARCHAR(10),  
    email VARCHAR(150) NOT NULL UNIQUE  
);
```

## Admin Table

```
29 • CREATE TABLE admins (  
30     admin_id INT PRIMARY KEY,  
31     first_name VARCHAR(100) NOT NULL,  
32     last_name VARCHAR(100),  
33     email VARCHAR(100) NOT NULL UNIQUE,  
34     phone VARCHAR(20),  
35     password VARCHAR(255) NOT NULL,  
36     profile_image VARCHAR(500),  
37     role ENUM('ADMIN', 'MANAGER', 'STAFF') NOT NULL DEFAULT 'STAFF',  
38     is_active BOOLEAN NOT NULL DEFAULT TRUE,  
39     last_login DATETIME DEFAULT NULL,  
40     created_at DATETIME NOT NULL,  
41     updated_at DATETIME NOT NULL  
42 );  
--
```

## Program Management (IT24101515)

### University Table

```
55 • CREATE TABLE universities (  
56     university_id INT PRIMARY KEY,  
57     university_name VARCHAR(200) NOT NULL,  
58     website_url VARCHAR(255),  
59     created_at DATETIME DEFAULT CURRENT_TIMESTAMP,  
60     updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP  
61 );  
62
```

### Program Table

```
73  
74 • CREATE TABLE programs (  
75     program_id INT PRIMARY KEY,  
76     university_id INT NOT NULL,  
77     program_name VARCHAR(255) NOT NULL,  
78     degree_type ENUM('BACHELOR', 'MASTER', 'PHD') NOT NULL,  
79     duration_years INT NOT NULL,  
80     description TEXT,  
81     tuition_fee_usd DECIMAL(10,2),  
82     status ENUM('ACTIVE', 'INACTIVE') NOT NULL DEFAULT 'ACTIVE',  
83     created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
84     updated_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
85     CONSTRAINT fk_university FOREIGN KEY (university_id)  
86         REFERENCES universities(university_id)  
87         ON DELETE CASCADE  
88 );  
89
```

## Application and Contact Management (IT24101636)

### Application Table

```
116 CREATE TABLE applications (  
117     application_id INT PRIMARY KEY,  
118     student_id INT NOT NULL,  
119     university_id INT NOT NULL,  
120     program_id INT NOT NULL,  
121     first_name VARCHAR(100) NOT NULL,  
122     last_name VARCHAR(100) NOT NULL,  
123     email VARCHAR(150) NOT NULL,  
124     phone_number VARCHAR(20),  
125     gender VARCHAR(10),  
126     date_of_birth DATE,  
127     application_type ENUM('BACHELOR', 'SPECIALIST', 'MASTER', 'PHD') NOT NULL,  
128     cv_file_path VARCHAR(500),  
129     application_date DATETIME NOT NULL,  
130     admin_id INT,  
131     application_status ENUM('PENDING', 'APPROVED', 'REJECTED') NOT NULL DEFAULT 'PENDING',  
132     created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
133     updated_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
134  
135     CONSTRAINT fk_application_student FOREIGN KEY (student_id)  
136         REFERENCES students(student_id) ON DELETE CASCADE,  
137     CONSTRAINT fk_application_university FOREIGN KEY (university_id)  
138         REFERENCES universities(university_id) ON DELETE CASCADE,  
139     CONSTRAINT fk_application_program FOREIGN KEY (program_id)  
140         REFERENCES programs(program_id) ON DELETE CASCADE,  
141     CONSTRAINT fk_application_admin FOREIGN KEY (admin_id)  
142         REFERENCES admins(admin_id) ON DELETE SET NULL  
143 );  
144
```

### Contact Message Table

```
116 CREATE TABLE contact_messages (  
117     message_id INT PRIMARY KEY,  
118     student_id INT NULL,  
119     program_id INT NULL,  
120     admin_id INT NULL,  
121     name VARCHAR(100) NOT NULL,  
122     email VARCHAR(150) NOT NULL,  
123     message_type ENUM('PROGRAM_INFO', 'SCHOLARSHIP_INFO', 'OTHER') NOT NULL,  
124     message_content TEXT NOT NULL,  
125     response_content TEXT,  
126     created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
127     updated_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
128  
129     CONSTRAINT fk_message_student FOREIGN KEY (student_id)  
130         REFERENCES students(student_id) ON DELETE SET NULL,  
131     CONSTRAINT fk_message_program FOREIGN KEY (program_id)  
132         REFERENCES programs(program_id) ON DELETE SET NULL,  
133     CONSTRAINT fk_message_admin FOREIGN KEY (admin_id)  
134         REFERENCES admins(admin_id) ON DELETE SET NULL  
135 );
```

## Payment and Notification Management (IT24101493)

Payment Table

Notification Table

```
• CREATE TABLE notification_logs (  
    notification_id INT PRIMARY KEY,  
    student_id INT NOT NULL,  
    type ENUM('APPLICATION_UPDATE', 'PAYMENT_CONFIRMATION', 'GENERAL') NOT NULL,  
    message TEXT NOT NULL,  
    subject VARCHAR(255),  
    recipient_email VARCHAR(150),  
    recipient_phone VARCHAR(20),  
    status ENUM('PENDING', 'SENT', 'DELIVERED', 'FAILED') NOT NULL DEFAULT 'PENDING',  
    sent_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    delivered_at DATETIME NULL,  
    failure_reason TEXT,  
    CONSTRAINT fk_notification_student FOREIGN KEY (student_id)  
        REFERENCES students(student_id) ON DELETE CASCADE  
);
```

## 4.2 Populating Data

### User and Admin Management (IT24101493)

Insert Data into the admin table

```
18
19 • INSERT INTO students
20 (student_id, first_name, last_name, date_of_birth, current_address, phone_number, gender, email)
21 VALUES
22 (1, 'Nimal', 'Perera', '2001-03-12', '135/3 Kandy', '0712345678', 'MALE', 'nimal.perera@gmail.com'),
23 (2, 'Tharushi', 'Senanayaka', '2002-08-05', 'No.10 Colombo', '0779876543', 'FEMALE', 'tharushi.s@gmail.com'),
24 (3, 'Sunil', 'Fernando', '2000-06-22', '25 Galle Road, Colombo', '0756547890', 'MALE', 'sunil.fernando@gmail.com'),
25 (4, 'Chamodi', 'Wijesinghe', '2003-01-15', '42 Matara', '0763456789', 'FEMALE', 'chamodi.w@gmail.com'),
26 (5, 'Ravindu', 'De Silva', '2001-09-09', '12 Kurunegala', '0722345678', 'MALE', 'ravindu.dsilva@gmail.com');
27
```

Insert Data into the admin table

```
44
45 • INSERT INTO admins
46 (admin_id, first_name, last_name, email, phone, password, profile_image, role, is_active, last_login, created_at, updated_at)
47 VALUES
48 (1, 'Subash', 'Mannapperuma', 'subash.m@gmail.com', '0712345678', 'Subash678', 'uploads/admins/subash.jpg', 'ADMIN', TRUE, NOW(), NOW(), NOW()),
49 (2, 'Imanka', 'Wimalasiri', 'imanka.w@gmail.com', '0774567890', 'Imanka890', 'uploads/admins/imanka.png', 'MANAGER', TRUE, NOW(), NOW(), NOW()),
50 (3, 'Thanuja', 'Mallikarachchi', 'thanuja.m@gmail.com', '0759876543', 'Thanuja543', 'uploads/admins/thanuja.jpg', 'MANAGER', TRUE, NOW(), NOW(), NOW()),
51 (4, 'Nadeesha', 'Perera', 'nadeesha.p@gmail.com', '0761122334', 'Nadeesha334', 'uploads/admins/nadeesha.png', 'STAFF', TRUE, NOW(), NOW(), NOW()),
52 (5, 'Ravindu', 'Fernando', 'ravindu.f@gmail.com', '0752233445', 'Ravindu445', 'uploads/admins/ravindu.jpg', 'STAFF', TRUE, NOW(), NOW(), NOW());
53
```

## Program Management (IT24101515)

Insert Data into the university table

```
• INSERT INTO universities (university_id, university_name, website_url, created_at, updated_at)
VALUES
(1, 'Chuvash State Pedagogical University', 'https://www.chgpu.edu.ru', NOW(), NOW()),
(2, 'Samara National Research University', 'https://www.ssau.ru', NOW(), NOW()),
(3, 'Yaroslavl State Technical University (YSTU)', 'https://www.ystu.ru', NOW(), NOW()),
(4, 'Chuvash State Agrarian University', 'https://www.chuvsu.ru', NOW(), NOW()),
(5, 'Lobachevsky State University of Nizhny Novgorod (UNN)', 'https://www.unn.ru', NOW(), NOW()),
(6, 'Kazan Innovative University', 'https://www.ikiu.ru', NOW(), NOW());
```

Insert Data into the program table

```
• # 1 Chuvash State Pedagogical University
INSERT INTO programs (program_id, university_id, program_name, degree_type, duration_years, description, tuition_fee_usd, status, created_at, updated_at)
VALUES
(1, 1, 'Informatics and Computer Engineering', 'BACHELOR', 4, 'Computer Science & IT Department', 8000, 'ACTIVE', NOW(), NOW());
# 2 Samara National Research University
INSERT INTO programs (program_id, university_id, program_name, degree_type, duration_years, description, tuition_fee_usd, status, created_at, updated_at)
VALUES
(3, 2, 'Aerospace Engineering', 'BACHELOR', 4, 'Engineering & Aerospace Department', 10000, 'ACTIVE', NOW(), NOW());
# 3 Yaroslavl State Technical University (YSTU)
INSERT INTO programs (program_id, university_id, program_name, degree_type, duration_years, description, tuition_fee_usd, status, created_at, updated_at)
VALUES
(7, 3, 'Software Engineering', 'BACHELOR', 4, 'IT & Software Department', 8000, 'ACTIVE', NOW(), NOW());
# 4 Chuvash State Agrarian University
INSERT INTO programs (program_id, university_id, program_name, degree_type, duration_years, description, tuition_fee_usd, status, created_at, updated_at)
VALUES
(8, 4, 'Biotechnology and Agronomy', 'BACHELOR', 4, 'Agriculture & Life Sciences Department', 7500, 'ACTIVE', NOW(), NOW());
# 5 Lobachevsky State University of Nizhny Novgorod (UNN)
INSERT INTO programs (program_id, university_id, program_name, degree_type, duration_years, description, tuition_fee_usd, status, created_at, updated_at)
VALUES
(10, 5, 'Biology and Biomedicine', 'BACHELOR', 4, 'Science & Medicine Department', 8500, 'ACTIVE', NOW(), NOW());
# 6 Kazan Innovative University
INSERT INTO programs (program_id, university_id, program_name, degree_type, duration_years, description, tuition_fee_usd, status, created_at, updated_at)
VALUES
(13, 6, 'Information Technology and Security', 'BACHELOR', 4, 'IT & Security Department', 8000, 'ACTIVE', NOW(), NOW());
```

## Application and Contact Management (IT24101636)

### Insert Data into the application table

```
INSERT INTO applications
(application_id, student_id, university_id, program_id, first_name, last_name, email, phone_number, gender, date_of_birth, application_type,
cv_file_path, application_date, admin_id, application_status, created_at, updated_at)
VALUES
(1, 1, 3, 5, 'Nimal', 'Perera', 'nimal.perera@gmail.com', '0712345678', 'MALE', '2001-03-12', 'BACHELOR', 'uploads/cv/nimal3.pdf', NOW(), 2, 'PENDING', NOW(), NOW()),
(2, 2, 5, 9, 'Tharushi', 'Senanayaka', 'tharushi.s@gmail.com', '0779876543', 'FEMALE', '2002-08-05', 'BACHELOR', 'uploads/cv/tharushi3.pdf', NOW(), 3, 'APPROVED', NOW(), NOW()),
(4, 4, 6, 11, 'Chamodi', 'Wijesinghe', 'chamodi.w@gmail.com', '0763456789', 'FEMALE', '2003-01-15', 'BACHELOR', 'uploads/cv/chamodi2.pdf', NOW(), 4, 'PENDING', NOW(), NOW());
```

### Insert Data into the contact message table

```
175 • INSERT INTO contact_messages
176 (message_id, student_id, program_id, admin_id, name, email, message_type, message_content, response_content, created_at, updated_at)
177 VALUES
178 #Registered student asking about program details
179 (1, 1, 5, 2, 'Nimal Perera', 'nimal.perera@gmail.com', 'PROGRAM_INFO',
180 'Can you please provide more details about the Bachelor program at ABC University?',
181 'Sure! The Bachelor program at ABC University is a 4-year course with various scholarship options available.', NOW(), NOW()),
182
183 #Registered student asking about scholarships
184 (2, 2, 9, 3, 'Tharushi Senanayaka', 'tharushi.s@gmail.com', 'SCHOLARSHIP_INFO',
185 'Hi, are there any scholarships available for the Bachelor of IT program?',
186 'Yes, there are merit-based and need-based scholarships. Please submit your latest transcript to check eligibility.', NOW(), NOW());
187
```

## Payment and Notification Management (IT24101493)

Insert Data into the payment table

```
194 • INSERT INTO payments
195 (payment_id, application_id, slip_file, uploaded_at, approved_at, rejection_reason, notes, status)
196 VALUES
197 # 1. Payment uploaded by Nimal (still pending)
198 (1, 1, 'uploads/payments/slip_nimal1.jpg', NOW(), NULL, NULL, 'Awaiting admin approval', 'PENDING'),
199
200 # 2. Payment approved for Tharushi's application
201 (2, 2, 'uploads/payments/slip_tharushi1.jpg', NOW(), NOW(), NULL, 'Payment verified successfully', 'APPROVED');
202
```

Insert Data into the notification table

```
• INSERT INTO notification_logs
(notification_id, student_id, type, message, subject, recipient_email, recipient_phone, status, sent_at, delivered_at, failure_reason)
VALUES
# Application received message to Nimal
(1, 1, 'APPLICATION_UPDATE',
'Your application for Software Engineering has been received successfully.',
'Application Received',
'nimal.perera@gmail.com',
'0712345678',
'DELIVERED', NOW(), NOW(), NULL),

# Payment confirmation to Tharushi
(2, 2, 'PAYMENT_CONFIRMATION',
'Your payment has been verified and approved.',
'Payment Approved',
'tharushi.s@gmail.com',
'0779876543',
'DELIVERED', NOW(), NOW(), NULL);
```



## 4.3 Display Tables

### User and Admin Management (IT24101499)

#### Students Table

```
259
260 -- Get all students
261 • SELECT * FROM students;
```

Result Grid								
Filter Rows:		Edit:		Export/Import:		Wrap Cell Content: <a href="#">IA</a>		
student_id	first_name	last_name	date_of_birth	current_address	phone_number	gender	email	
1	Nimal	Perera	2001-03-12	135/3 Kandy	0771112223	MALE	nimal.perera@gmail.com	
2	Tharushi	Senanayaka	2002-08-05	No.10 Colombo	0779876543	FEMALE	tharushi.s@gmail.com	
3	Sunil	Fernando	2000-06-22	25 Galle Road, Colombo	0756547890	MALE	sunil.fernando@gmail.com	
4	Chamodi	Wijesinghe	2003-01-15	42 Matara	0763456789	FEMALE	chamodi.w@gmail.com	
5	Ravindu	De Silva	2001-09-09	12 Kurunegala	0722345678	MALE	ravindu.dsilva@gmail.com	
* NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

#### Admin Table

Result Grid												
Filter Rows:		Edit:		Export/Import:		Wrap Cell Content: <a href="#">IA</a>						
admin_id	first_name	last_name	email	phone	password	profile_image	role	is_active	last_login	created_at	updated_at	
1	Subash	Mannapperuma	subash.m@gmail.com	0712345678	Subash678	uploads/admins/subash.jpg	ADMIN	1	2025-10-10 12:06:50	2025-10-10 12:06:50	2025-10-10 12:06:50	
2	Imanka	Wimalasiri	imanka.w@gmail.com	0774567890	Imanka890	uploads/admins/imanka.png	MANAGER	1	2025-10-10 12:06:50	2025-10-10 12:06:50	2025-10-10 12:06:50	
3	Thanuja	Malikarachchi	thanuja.m@gmail.com	0759876543	Thanuja543	uploads/admins/thanuja.jpg	MANAGER	1	2025-10-10 12:06:50	2025-10-10 12:06:50	2025-10-10 12:06:50	
4	Nadeesha	Perera	nadeesha.p@gmail.com	0761122334	Nadeesha334	uploads/admins/nadeesha.png	STAFF	1	2025-10-10 12:06:50	2025-10-10 12:06:50	2025-10-10 12:06:50	
5	Ravindu	Fernando	ravindu.f@gmail.com	0752233445	Ravindu445	uploads/admins/ravindu.jpg	STAFF	1	2025-10-10 12:06:50	2025-10-10 12:06:50	2025-10-10 12:06:50	
* NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

## Program Management (IT24101515)

### University Table

Result Grid					
Filter Rows: <input type="text"/>					
Edit:					
Export/Import:					
Wrap Cell Content:					
	university_id	university_name	website_url	created_at	updated_at
▶	1	Chuvash State Pedagogical University	https://www.chgpu.edu.ru	2025-10-10 12:24:20	2025-10-10 12:24:20
	2	Samara National Research University	https://www.ssau.ru	2025-10-10 12:24:20	2025-10-10 12:24:20
	3	Yaroslavl State Technical University (YSTU)	https://www.ystu.ru	2025-10-10 12:24:20	2025-10-10 12:24:20
	4	Chuvash State Agrarian University	https://www.chuvsu.ru	2025-10-10 12:24:20	2025-10-10 12:24:20
	5	Lobachevsky State University of Nizhny Novgor...	https://www.unn.ru	2025-10-10 12:24:20	2025-10-10 12:24:20
	6	Kazan Innovative University	https://www.ikiu.ru	2025-10-10 12:24:20	2025-10-10 12:24:20
•	NULL	NULL	NULL	NULL	NULL

### Program Table

```
305 -- Get all programs
306 • SELECT * FROM programs;
307
```

Result Grid										
Filter Rows:			Edit:		Export/Import:		Wrap Cell Content:			
program_id	university_id	program_name	degree_type	duration_years	description	tuition_fee_usd	status	created_at	updated_at	
1	1	Informatics and Computer Engineering	BACHELOR	4	Computer Science & IT Department	8000.00	ACTIVE	2025-10-10 15:31:31	2025-10-10 15:31:31	
3	2	Aerospace Engineering	BACHELOR	4	Engineering & Aerospace Department	10000.00	ACTIVE	2025-10-10 15:31:31	2025-10-10 15:31:31	
7	3	Software Engineering	BACHELOR	4	IT & Software Department	8000.00	ACTIVE	2025-10-10 15:31:31	2025-10-10 15:31:31	
8	4	Biotechnology and Agronomy	BACHELOR	4	Agriculture & Life Sciences Department	7500.00	ACTIVE	2025-10-10 15:31:31	2025-10-10 15:31:31	
10	5	Biology and Biomedicine	BACHELOR	4	Science & Medicine Department	8500.00	ACTIVE	2025-10-10 15:31:31	2025-10-10 15:31:31	
13	6	Information Technology and Security	BACHELOR	4	IT & Security Department	8000.00	ACTIVE	2025-10-10 15:31:31	2025-10-10 15:31:31	
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

## Application and Contact Management (IT24101636)

### Applications Table

application_id	student_id	university_id	program_id	first_name	last_name	email	phone_number	gender	date_of_birth	application_type	cv_file_path	application_date	admin_id	application_status	created_at
1	1	3	7	Nimal	Perera	nimal.perera@gmail.com	0712345678	MALE	2001-03-12	BACHELOR	uploads/cv/nimal3.pdf	2025-10-10 15:34:55	2	PENDING	2025-10-10 15:34:55
2	2	5	10	Tharushi	Senanayaka	tharushi.s@gmail.com	0779876543	FEMALE	2002-08-05	BACHELOR	uploads/cv/tharushi3.pdf	2025-10-10 15:34:55	3	APPROVED	2025-10-10 15:34:55
3	4	6	13	Chamodi	Wijesinghe	chamodi.w@gmail.com	0763456789	FEMALE	2003-01-15	BACHELOR	uploads/cv/chamodi2.pdf	2025-10-10 15:34:55	4	PENDING	2025-10-10 15:34:55

### Contact message table

```
344 -- Get all messages
345 SELECT * FROM contact_messages;
```

message_id	student_id	program_id	admin_id	name	email	message_type	message_content	response_content	created_at	updated_at
1	1	7	2	Nimal Perera	nimal.perera@gmail.com	PROGRAM_INFO	Can you please provide more details about the ...	Sure! The Bachelor program at ABC University i...	2025-10-10 15:35:58	2025-10-10 15:35:58
2	2	10	3	Tharushi Senanayaka	tharushi.s@gmail.com	SCHOLARSHIP_INFO	Hi, are there any scholarships available for the ...	Yes, there are merit-based and need-based sch...	2025-10-10 15:35:58	2025-10-10 15:35:58

## Payment and Notification Management (IT24101493)

### Payment Table

```
362 -- Get all payments
363 • SELECT * FROM payments;
```

payment_id	application_id	slip_file	uploaded_at	approved_at	rejection_reason	notes	status
1	1	uploads/payments/slip_nimal1.jpg	2025-10-10 15:36:07	NULL	NULL	Awaiting admin approval	PENDING
2	2	uploads/payments/slip_tharushi1.jpg	2025-10-10 15:36:07	2025-10-10 15:36:07	NULL	Payment verified successfully	APPROVED
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### Notification Table

```
380 -- Get all notifications
381 • SELECT * FROM notification_logs;
```

notification_id	student_id	type	message	subject	recipient_email	recipient_phone	status	sent_at	delivered_at	failure_reason
1	1	APPLICATION_UPDATE	Your application for Software Engineering has b...	Application Received	nimal.perera@gmail.com	0712345678	DELIVERED	2025-10-10 15:36:17	2025-10-10 15:36:17	NULL
2	2	PAYMENT_CONFIRMATION	Your payment has been verified and approved.	Payment Approved	tharushi.s@gmail.com	0779876543	DELIVERED	2025-10-10 15:36:17	2025-10-10 15:36:17	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## 4.4 SQL Queries

### User and Admin Management (IT24101493)

- 1) Get only student first name and last name

```
263 -- Get only student first name and last name
264 • SELECT first_name, last_name FROM students;
```

first_name	last_name
Nimal	Perera
Tharushi	Senanayaka
Sunil	Fernando
Chamodi	Wijesinghe
Ravindu	De Silva

- 2) Find students born after 2001-01-01

```
269 -- Find students born after 2001-01-01
270 • SELECT * FROM students WHERE date_of_birth > '2001-01-01';
271
```

student_id	first_name	last_name	date_of_birth	current_address	phone_number	gender	email
1	Nimal	Perera	2001-03-12	135/3 Kandy	0771112223	MALE	nimal.perera@gmail.com
2	Tharushi	Senanayaka	2002-08-05	No. 10 Colombo	0779876543	FEMALE	tharushi.s@gmail.com
4	Chamodi	Wijesinghe	2003-01-15	42 Matara	0763456789	FEMALE	chamodi.w@gmail.com
5	Ravindu	De Silva	2001-09-09	12 Kurunegala	0722345678	MALE	ravindu.dsilva@gmail.com
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- 3) Update a student's phone number

```
-- Update a student's phone number
• UPDATE students
  SET phone_number = '0771112223'
  WHERE student_id = 1;
```

student_id	first_name	last_name	date_of_birth	current_address	phone_number	gender	email
1	Nimal	Perera	2001-03-12	135/3 Kandy	0771112223	MALE	nimal.perera@gmail.com

4) Delete a student by ID

```
-- Delete a student by ID
```

```
DELETE FROM students WHERE student_id = 3;
```

student_id	first_name	last_name	date_of_birth	current_address	phone_number	gender	email
1	Nimal	Perera	2001-03-12	135/3 Kandy	0771112223	MALE	nimal.perera@gmail.com
2	Tharushi	Senanayaka	2002-08-05	No. 10 Colombo	0779876543	FEMALE	tharushi.s@gmail.com
4	Chamodi	Wijesinghe	2003-01-15	42 Matara	0763456789	FEMALE	chamodi.w@gmail.com
5	Ravindu	De Silva	2001-09-09	12 Kurunegala	0722345678	MALE	ravindu.dsilva@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

5) Get admin names with roles

```
290 -- Get admin names with roles
```

```
291 • SELECT first_name, last_name, role FROM admins;
```

```
292
```

first_name	last_name	role
Subash	Mannapperuma	ADMIN
Imanka	Wimalasiri	MANAGER
Thanuja	Mallikarachchi	MANAGER
Nadeesha	Perera	STAFF
Ravindu	Fernando	STAFF

6) Count number of admins per role

```
293 -- Count number of admins per role
```

```
294 • SELECT role, COUNT(*) AS total_admins
```

```
295 FROM admins
```

```
296 GROUP BY role;
```

```
297
```

```
298
```

role	total_admins
ADMIN	1
MANAGER	2
STAFF	2

## Program Management (IT24101515)

1) Get only university names and websites

```
318 -- Get only university names and websites
319 • SELECT university_name, website_url FROM universities;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
university_name	website_url		
▶ Chuvash State Pedagogical University	https://www.chgpu.edu.ru		
Samara National Research University	https://www.ssau.ru		
Yaroslavl State Technical University (YSTU)	https://www.ystu.ru		
Chuvash State Agrarian University	https://www.chuvsu.ru		
Lobachevsky State University of Nizhny Novgor...	https://www.unn.ru		
Kazan Innovative University	https://www.ikiu.ru		

2) Sort universities alphabetically

```
301
302 -- Sort universities alphabetically
303 • SELECT * FROM universities ORDER BY university_name ASC;
304
```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Contents:
university_id	university_name	website_url	created_at	updated_at
▶ 4	Chuvash State Agrarian University	https://www.chuvsu.ru	2025-10-10 15:31:21	2025-10-10 15:31:21
1	Chuvash State Pedagogical University	https://www.chgpu.edu.ru	2025-10-10 15:31:21	2025-10-10 15:31:21
6	Kazan Innovative University	https://www.ikiu.ru	2025-10-10 15:31:21	2025-10-10 15:31:21
5	Lobachevsky State University of Nizhny Novgor...	https://www.unn.ru	2025-10-10 15:31:21	2025-10-10 15:31:21
2	Samara National Research University	https://www.ssau.ru	2025-10-10 15:31:21	2025-10-10 15:31:21
3	Yaroslavl State Technical University (YSTU)	https://www.ystu.ru	2025-10-10 15:31:21	2025-10-10 15:31:21
• NULL	NULL	NULL	NULL	NULL

3) Sort programs by tuition fee descending

```
314 -- Sort programs by tuition fee descending
315 • SELECT program_name, tuition_fee_usd
316 FROM programs
317 ORDER BY tuition_fee_usd DESC;
318
319 -- ADDITION TABLE
```

Result Grid	Filter Rows:	Export:	W
program_name	tuition_fee_usd		
▶ Aerospace Engineering	10000.00		
Biology and Biomedicine	8500.00		
Informatics and Computer Engineering	8000.00		
Software Engineering	8000.00		
Information Technology and Security	8000.00		
Biotechnology and Agronomy	7500.00		

4) Count programs per degree type

```
310
311 -- Count programs per degree type
312 • SELECT degree_type, COUNT(*) AS total_programs
313 FROM programs
314 GROUP BY degree_type;
315
316
```

Result Grid | Filter Rows:  | Export: | Wrap Cell

degree_type	total_programs
BACHELOR	6

5) Get programs with tuition fees less than 8000

```
279
280 -- Get programs with tuition fee less than 8000
281 • SELECT program_name, tuition_fee_usd FROM programs WHERE tuition_fee_usd < 8000;
282
283
284
```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content:

program_name	tuition_fee_usd
Biotechnology and Agronomy	7500.00



## Application and Contact Management (IT24101636)

### 1) Count applications per status

```
328 -- Count applications per status
329 • SELECT application_status, COUNT(*) AS total
330 FROM applications
331 GROUP BY application_status;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Co

application_status	total
PENDING	2
APPROVED	1

### 2) Find applications for a specific university

```
333 -- Find applications for a specific university
334 • SELECT * FROM applications WHERE university_id = 3;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

application_id	student_id	university_id	program_id	first_name	last_name	email	phone_number	gender	date_of_birth	application_type	cv_file_path	application_date	admin_id	application_status	created_at
1	1	3	7	Nimal	Perera	nimal.perera@gmail.com	0712345678	MALE	2001-03-12	BACHELOR	uploads/cv/nimal3.pdf	2025-10-10 15:34:55	2	PENDING	2025-10-10 15:34:55

### 3) List applications with student first and last name

```
324 -- List applications with student first and last name
325 • SELECT a.application_id, s.first_name, s.last_name, a.application_status
326 FROM applications a
327 JOIN students s ON a.student_id = s.student_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

application_id	first_name	last_name	application_status
1	Nimal	Perera	PENDING
2	Tharushi	Senanayaka	APPROVED
3	Chamodi	Wijesinghe	APPROVED

#### 4) Find messages by type PROGRAM\_INFO

```
346
347 -- Find messages by type PROGRAM_INFO
348 • SELECT * FROM contact_messages WHERE message_type = 'PROGRAM_INFO';
349
350 -- Count messages per type
351 • SELECT message_type, COUNT(*) AS total
```

message_id	student_id	program_id	admin_id	name	email	message_type	message_content	response_content	created_at	updated_at
1	1	7	2	Nimal Perera	nimal.perera@gmail.com	PROGRAM_INFO	Can you please provide more details about the ...	Sure! The Bachelor program at ABC University i...	2025-10-10 15:35:58	2025-10-10 15:35:58

#### 5) Count messages per type

```
350 -- Count messages per type
351 • SELECT message_type, COUNT(*) AS total
352 FROM contact_messages
353 GROUP BY message_type;
354
355 show messages with student info
```

message_type	total
PROGRAM_INFO	1
SCHOLARSHIP_INFO	1

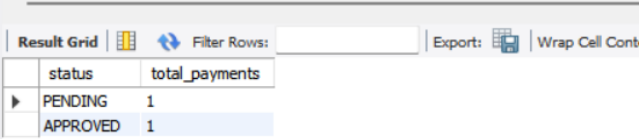
#### 6) Update response content of a message

```
-- Update response content of a message
• UPDATE contact_messages
  SET response_content = 'We have sent you the program details.'
  WHERE message_id = 2;
```

## Payment and Notification Management (IT24101493)

### 1) Count payments by status

```
366 -- Count payments by status
367 • SELECT status, COUNT(*) AS total_payments
368 FROM payments
369 GROUP BY status;
370
371
```



Result Grid | Filter Rows: | Export: | Wrap Cell Cont

status	total_payments
PENDING	1
APPROVED	1

### 2) Approving payment

```
72 -- Approve a payment
73 • UPDATE payments
74 SET status = 'APPROVED', approved_at = NOW()
75 WHERE payment_id = 1;
```

### 3) Delete rejected payments

```
-- Delete rejected payments
• DELETE FROM payments WHERE status = 'REJECTED';
```

4) Count notifications by type

```
385 -- Count notifications by type
386 • SELECT type, COUNT(*) AS total_notifications
387 FROM notification_logs
388 GROUP BY type;
389
390
```

Result Grid | Filter Rows: | Export: | Wrap

	type	total_notifications
▶	APPLICATION_UPDATE	1
	PAYMENT_CONFIRMATION	1

5) Update notification status to DELIVERED

```
-- Update notification status to DELIVERED
UPDATE notification_logs
SET status = 'DELIVERED', delivered_at = NOW()
WHERE notification_id = 1;
```

6) Delete failed notifications

```
-- Delete failed notifications
DELETE FROM notification_logs WHERE status = 'FAILED';
```

## **5 Conclusion**

In conclusion, the SIT International Student Management System successfully addresses the important operational needs of an education consultancy. Through the design and development of a normalized relational database, we have created a strong foundation for managing students, universities, programs, applications, payments, and communications. The system provides a clear and linear workflow for students applying for foreign education and administrators managing the entire process easily. This project significantly enriched our hands-on understanding of database design principles, from the very first steps of client requirements gathering through to the final creation of a complex, multi-module system. The database is expandable, secure, and ready for further enhancement and integration with front-end applications.