

Database Design

Sany3y

Version 1.0

Nov of 2025

Table of Contents

Team Members	3
Introduction	3
Database Objectives.....	3
Entities and Attributes.....	4
Relationships.....	6
ERD.....	7
Summary	7

Team Members

ID	Name	Email
21030343	Mohammed Atef	mohamed45452020@gmail.com
21060536	Steven Amin	stevenameen02@gmail.com
21040560	Omnia Ashraf	omniaamohamed102@gmail.com
21056226	Kenzi Shibl	kebzishibl@gmail.com

Introduction

This document outlines the database design for the Sany3y full-stack website. It defines entities, relationships, constraints, and data structures required to support the platform's operations.

Database Objectives

- Ensure data consistency and integrity.
- Support user, technician, booking, and admin operations.
- Provide efficient querying for dashboard analytics.
- Enable scalable growth for future platform expansion.

Entities and Attributes

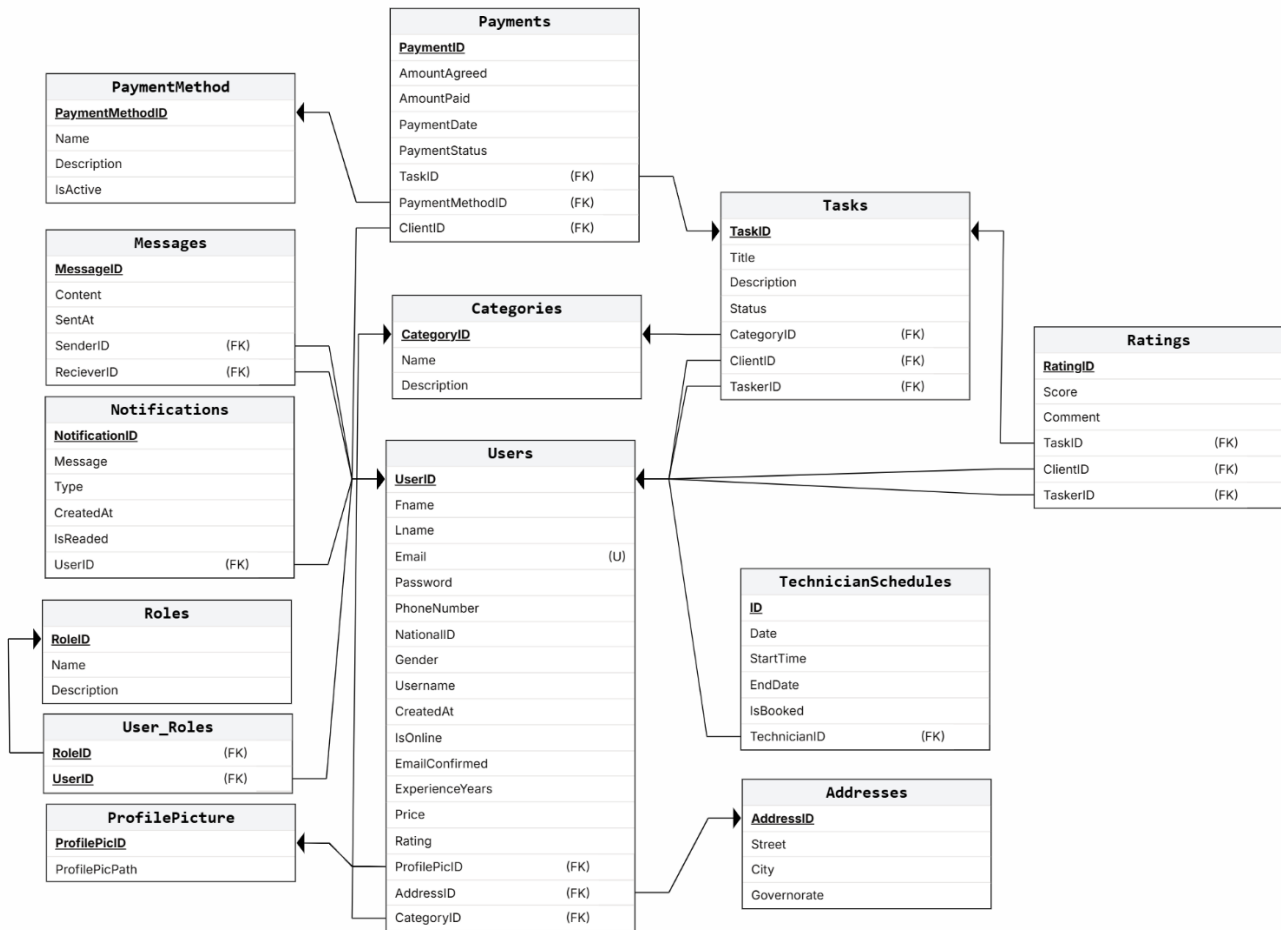
Entity	Description	Key Attributes
ProfilePicture	Stores the paths of users' profile pictures	<ul style="list-style-type: none"> • ProfilePicID (PK) • ProfilePicPath
Addresses	Stores address details (street, city, governorate)	<ul style="list-style-type: none"> • AddressID (PK) • Street • City • Governorate
Categories	Represents the categories of services or skills in the system	<ul style="list-style-type: none"> • CategoryID (PK) • Name • Description
Users	Stores information about system users (clients/technicians)	<ul style="list-style-type: none"> • UserID (PK) • Email (Unique) • Fname • Lname • Password • PhoneNumber • NationalID • Gender • Username • CreatedAt • ProfilePicID (FK) • AddressID (FK) • CategoryID (FK)
Tasks	Tasks created between clients and technicians	<ul style="list-style-type: none"> • TaskID (PK) • Title • Description • Status • CategoryID (FK) • ClientID (FK) • TaskerID (FK)
Notifications	Notifications sent to users	<ul style="list-style-type: none"> • NotificationID (PK) • Message • Type • CreatedAt • IsReaded • UserID (FK)

Messages	Messages exchanged between clients and technicians	<ul style="list-style-type: none"> • MessageID (PK) • Content • SentAt • SenderID (FK) • ReceiverID (FK)
Ratings	Ratings given by clients to technicians after completing tasks	<ul style="list-style-type: none"> • RatingID (PK) • Score • Comment • TaskID (FK) • ClientID (FK) • TaskerID (FK)
PaymentMethod	Available payment methods in the system	<ul style="list-style-type: none"> • PaymentMethodID (PK) • Name • Description • IsActive
Payments	Records payments related to tasks	<ul style="list-style-type: none"> • PaymentID (PK) • AmountAgreed • AmountPaid • PaymentDate • PaymentStatus • TaskID (FK) • PaymentMethodID (FK) • ClientID (FK)
Roles	User roles (Admin / Client / Technician)	<ul style="list-style-type: none"> • RoleID (PK) • Name • Description
User_Roles	Many-to-many relationship between users and roles	<ul style="list-style-type: none"> • RoleID + UserID (Composite PK) • RoleID (FK) • UserID (FK)
TechnicianSchedules	Schedules for technicians showing availability	<ul style="list-style-type: none"> • ID (PK) • Date • StartTime • EndDate • IsBooked • TechnicianID (FK)

Relationships

Relationship	Type	Description
ProfilePicture → Users	One-to-Many	Each user has one profile picture, but a profile picture could theoretically be linked to multiple users (depending on your design, usually 1-to-1).
Addresses → Users	One-to-Many	Each user has one address, but an address can belong to multiple users.
Categories → Users	One-to-Many	Each user belongs to one category (e.g., skill), each category can have many users.
Categories → Tasks	One-to-Many	Each task belongs to a category; a category can have many tasks.
Users → Tasks (Client)	One-to-Many	Each client can create many tasks; each task has one client.
Users → Tasks (Tasker/Technician)	One-to-Many	Each technician can be assigned to many tasks; each task has one technician.
Users → Notifications	One-to-Many	Each user can receive many notifications.
Users → Messages (Sender)	One-to-Many	Each user can send many messages.
Users → Messages (Receiver)	One-to-Many	Each user can receive many messages.
Tasks → Ratings	One-to-Many	Each task can have ratings.
Users → Ratings (Client)	One-to-Many	Each client can rate multiple tasks.
Users → Ratings (Tasker)	One-to-Many	Each technician can be rated multiple times.
PaymentMethod → Payments	One-to-Many	Each payment method can be used for many payments.
Tasks → Payments	One-to-Many	Each task can have many payments.
Users → Payments (Client)	One-to-Many	Each client can make many payments.
Roles → User_Roles	One-to-Many	Each role can be assigned to many users.
Users → User_Roles	One-to-Many	Each user can have multiple roles.
Users → TechnicianSchedules	One-to-Many	Each technician can have multiple schedule entries.

ERD



Summary

This database design supports the core functionalities of the Sany3y platform, ensures data integrity, allows scalability, and maintains efficient query performance.