```sql
-- Create a new database called 'freelancer'
-- Connect to the 'master' database to run this snippet
USE master
GO
IF EXISTS (SELECT name FROM sys.databases WHERE name = 'freelancer')
BEGIN
    DROP DATABASE freelancer;
END

CREATE DATABASE freelancer;

USE freelancer
GO

DROP TABLE IF EXISTS dbo.ADMIN_USERS;

CREATE TABLE dbo.ADMIN_USERS (
    AdminID INT IDENTITY(1,1) PRIMARY KEY,
    FullName NVARCHAR(100) NOT NULL,
    Email NVARCHAR(100) NOT NULL UNIQUE,
    Admin_Role NVARCHAR(50) NOT NULL,
    AccessLevel INT CHECK (AccessLevel BETWEEN 1 AND 5) NOT NULL,
    CreatedAt DATETIME DEFAULT GETDATE()
);

DROP TABLE IF EXISTS dbo.REPORTS;
CREATE TABLE dbo.REPORTS (
    ReportID INT IDENTITY(1,1) PRIMARY KEY,
    ReportType NVARCHAR(50) NOT NULL,
    GenerationDate DATETIME DEFAULT GETDATE() NOT NULL,
    DownloadURL NVARCHAR(255) NOT NULL,
    AdminID INT NOT NULL,
    CONSTRAINT FK_Reports_Admin FOREIGN KEY (AdminID)
        REFERENCES dbo.ADMIN_USERS(AdminID) ON DELETE CASCADE ON UPDATE CASCADE
);

DROP TABLE IF EXISTS dbo.TEAMS;
DROP TABLE IF EXISTS dbo.FREELANCERS;
DROP TABLE IF EXISTS dbo.TEAMS_FREELANCERS;


CREATE TABLE dbo.TEAMS (
    TeamID INT IDENTITY(1,1) PRIMARY KEY,
    TeamRole NVARCHAR(50) NOT NULL,
    JoinDate DATETIME DEFAULT GETDATE() NOT NULL
);



CREATE TABLE dbo.FREELANCERS (
    FreelancerID INT IDENTITY(1,1) PRIMARY KEY,
    FirstName NVARCHAR(50) NOT NULL,
    LastName NVARCHAR(50) NOT NULL,
    Phone NVARCHAR(15) NOT NULL UNIQUE,
    ProfilePictureURL NVARCHAR(255),
    Skills NVARCHAR(500) NOT NULL,
    ExperienceLevel NVARCHAR(50) CHECK (ExperienceLevel IN ('Beginner',
'Intermediate', 'Expert')) NOT NULL,
    Rating DECIMAL(3,2) CHECK (Rating BETWEEN 0 AND 5) DEFAULT 0 NOT NULL,
    AvailabilityStatus NVARCHAR(50) CHECK (AvailabilityStatus IN ('Available',
'Busy', 'On Vacation')) NOT NULL,
    PortfolioURL NVARCHAR(255),
    TotalEarnings DECIMAL(15,2) DEFAULT 0 CHECK (TotalEarnings >= 0) NOT NULL,
    CreatedAt DATETIME DEFAULT GETDATE() NOT NULL
);

CREATE TABLE dbo.TEAMS_FREELANCERS (
    TeamID INT NOT NULL,
    FreelancerID INT NOT NULL,
    AssignedDate DATETIME DEFAULT GETDATE() NOT NULL,
    PRIMARY KEY (TeamID, FreelancerID),
    CONSTRAINT FK_TeamsFreelancers_Teams FOREIGN KEY (TeamID)
        REFERENCES dbo.TEAMS(TeamID) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT FK_TeamsFreelancers_Freelancers FOREIGN KEY (FreelancerID)
        REFERENCES dbo.FREELANCERS(FreelancerID) ON DELETE CASCADE ON UPDATE
CASCADE
);

DROP TABLE IF EXISTS dbo.CLIENTS;

CREATE TABLE dbo.CLIENTS (
    ClientID INT IDENTITY(1,1) PRIMARY KEY,
    CompanyName NVARCHAR(255) NOT NULL,
    FirstName NVARCHAR(50) NOT NULL,
    LastName NVARCHAR(50) NOT NULL,
    Email NVARCHAR(100) NOT NULL UNIQUE,
    Phone NVARCHAR(15) NOT NULL UNIQUE,
    CompanyWebsiteURL NVARCHAR(255),
    ClientRating DECIMAL(3,2) CHECK (ClientRating BETWEEN 0 AND 5) DEFAULT 0 NOT
NULL,
    TotalProjectsPosted INT DEFAULT 0 CHECK (TotalProjectsPosted >= 0) NOT NULL,
    IndustryType NVARCHAR(100) NOT NULL,
    CreatedAt DATETIME DEFAULT GETDATE() NOT NULL
);

DROP TABLE IF EXISTS dbo.SUBSCRIPTION_PLANS;

CREATE TABLE dbo.SUBSCRIPTION_PLANS (
    SubscriptionID INT IDENTITY(1,1) PRIMARY KEY,
    PlanType NVARCHAR(100) NOT NULL UNIQUE
);

DROP TABLE IF EXISTS dbo.CLIENT_SUBSCRIPTIONS;
DROP TABLE IF EXISTS dbo.FREELANCER_SUBSCRIPTIONS;

CREATE TABLE dbo.CLIENT_SUBSCRIPTIONS (
    ClientID INT NOT NULL,
    SubscriptionID INT NOT NULL,
    StartDate DATETIME DEFAULT GETDATE() NOT NULL,
    EndDate DATETIME NOT NULL,
    PaymentStatus NVARCHAR(50) CHECK (PaymentStatus IN ('Pending', 'Completed',
'Failed')) NOT NULL,
    PRIMARY KEY (ClientID, SubscriptionID),
    CONSTRAINT FK_Client_Subscriptions_Client FOREIGN KEY (ClientID)
        REFERENCES dbo.CLIENTS(ClientID) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT FK_Client_Subscriptions_Subscription FOREIGN KEY (SubscriptionID)
        REFERENCES dbo.SUBSCRIPTION_PLANS(SubscriptionID) ON DELETE CASCADE ON
UPDATE CASCADE
```

```
);

CREATE TABLE dbo.FREELANCER_SUBSCRIPTIONS (
    FreelancerID INT NOT NULL,
    SubscriptionID INT NOT NULL,
    StartDate DATETIME DEFAULT GETDATE() NOT NULL,
    EndDate DATETIME NOT NULL,
    PaymentStatus NVARCHAR(50) CHECK (PaymentStatus IN ('Pending', 'Completed',
'Failed')) NOT NULL,
    PRIMARY KEY (FreelancerID, SubscriptionID),
    CONSTRAINT FK_Freelancer_Subscriptions_Freelancer FOREIGN KEY (FreelancerID)
        REFERENCES dbo.FREELANCERS(FreelancerID) ON DELETE CASCADE ON UPDATE
CASCADE,
    CONSTRAINT FK_Freelancer_Subscriptions_Subscription FOREIGN KEY
(SubscriptionID)
        REFERENCES dbo.SUBSCRIPTION_PLANS(SubscriptionID) ON DELETE CASCADE ON
UPDATE CASCADE
);

DROP TABLE IF EXISTS dbo.DISPUTES;

CREATE TABLE dbo.DISPUTES (
    DisputeID INT IDENTITY(1,1) PRIMARY KEY,
    Reason NVARCHAR(255) NOT NULL,
    Status NVARCHAR(50) CHECK (Status IN ('Open', 'In Progress', 'Resolved',
'Closed')) NOT NULL,
    ResolutionDetails NVARCHAR(500),
    FreelancerID INT NOT NULL,
    ClientID INT NOT NULL,
    CreatedAt DATETIME DEFAULT GETDATE() NOT NULL,
    CONSTRAINT FK_Disputes_Freelancer FOREIGN KEY (FreelancerID)
        REFERENCES dbo.FREELANCERS(FreelancerID) ON DELETE CASCADE ON UPDATE
CASCADE,
    CONSTRAINT FK_Disputes_Client FOREIGN KEY (ClientID)
        REFERENCES dbo.CLIENTS(ClientID) ON DELETE CASCADE ON UPDATE CASCADE
);


DROP TABLE IF EXISTS dbo.PROJECTS;

CREATE TABLE dbo.PROJECTS (
    ProjectID INT IDENTITY(1,1) PRIMARY KEY,
    ProjectName NVARCHAR(255) NOT NULL,
    Description NVARCHAR(1000),
    Budget DECIMAL(15,2) CHECK (Budget >= 0) NOT NULL,
    Deadline DATETIME NOT NULL,
    SkillsRequired NVARCHAR(500),
    Status NVARCHAR(50) CHECK (Status IN ('Open', 'In Progress', 'Completed', 'On
Hold', 'Cancelled')) NOT NULL,
    TotalApplicants INT DEFAULT 0 CHECK (TotalApplicants >= 0) NOT NULL,
    PostedDate DATETIME DEFAULT GETDATE() NOT NULL,
    ClientID INT NOT NULL,
    TeamID INT NOT NULL,
    CONSTRAINT FK_Projects_Client FOREIGN KEY (ClientID)
        REFERENCES dbo.CLIENTS(ClientID) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT FK_Projects_Team FOREIGN KEY (TeamID)
        REFERENCES dbo.TEAMS(TeamID) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
DROP TABLE IF EXISTS dbo.EVENTS;
DROP TABLE IF EXISTS dbo.PAYMENTS;
DROP TABLE IF EXISTS dbo.MILESTONES;

CREATE TABLE dbo.MILESTONES (
    MilestoneID INT IDENTITY(1,1) PRIMARY KEY,
    MilestoneName NVARCHAR(255) NOT NULL,
    DueDate DATETIME NOT NULL,
    Status NVARCHAR(50) CHECK (Status IN ('Pending', 'Completed', 'In Progress'))
NOT NULL,
    ProjectID INT NOT NULL,
    EventID INT,
    CONSTRAINT FK_Milestones_Project FOREIGN KEY (ProjectID)
        REFERENCES dbo.PROJECTS(ProjectID) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE dbo.PAYMENTS (
    PaymentID INT IDENTITY(1,1) PRIMARY KEY,
    PaymentDate DATETIME NOT NULL,
    PaymentStatus NVARCHAR(50) CHECK (PaymentStatus IN ('Pending', 'Completed',
'Failed')) NOT NULL,
    Amount DECIMAL(15,2) NOT NULL,
    PaymentMethod NVARCHAR(50) NOT NULL,
    MilestoneID INT NOT NULL, -- Links to Milestones table
    CONSTRAINT FK_Payments_Milestone FOREIGN KEY (MilestoneID)
        REFERENCES dbo.MILESTONES(MilestoneID) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE dbo.EVENTS (
    EventID INT IDENTITY(1,1) PRIMARY KEY,
    EventName NVARCHAR(255) NOT NULL,
    TimeStamp DATETIME DEFAULT GETDATE() NOT NULL,
    Description NVARCHAR(1000),
    MilestoneID INT, -- Links to Milestones (SubType)
    PaymentID INT, -- Links to Payments (SubType)
    CONSTRAINT FK_Events_Milestone FOREIGN KEY (MilestoneID)
        REFERENCES dbo.MILESTONES(MilestoneID) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT FK_Events_Payment FOREIGN KEY (PaymentID)
        REFERENCES dbo.PAYMENTS(PaymentID) ON DELETE NO ACTION ON UPDATE NO ACTION
);

DROP TABLE IF EXISTS dbo.SERVICEFEES;

CREATE TABLE dbo.SERVICEFEES (
    ServiceFeeID INT IDENTITY(1,1) PRIMARY KEY,
    ServiceChargePercentage DECIMAL(5,2) CHECK (ServiceChargePercentage >= 0 AND
ServiceChargePercentage <= 100) NOT NULL,
    TotalFeeDeducted DECIMAL(15,2) CHECK (TotalFeeDeducted >= 0) NOT NULL,
    TransactionDate DATETIME DEFAULT GETDATE() NOT NULL,
    PaymentID INT NOT NULL, -- Links to Payments table
    CONSTRAINT FK_ServiceFees_Payment FOREIGN KEY (PaymentID)
        REFERENCES dbo.PAYMENTS(PaymentID) ON DELETE CASCADE ON UPDATE CASCADE
);

DROP TABLE IF EXISTS dbo.NOTIFICATIONS;

CREATE TABLE dbo.NOTIFICATIONS (
    NotificationID INT IDENTITY(1,1) PRIMARY KEY,
    Message NVARCHAR(1000) NOT NULL,
    Timestamp DATETIME DEFAULT GETDATE() NOT NULL,
    NotificationType NVARCHAR(50) NOT NULL,
    Status NVARCHAR(50) CHECK (Status IN ('Read', 'Unread')) NOT NULL,
    EventID INT, -- Links to Events table
    CONSTRAINT FK_Notifications_Event FOREIGN KEY (EventID)
        REFERENCES dbo.EVENTS(EventID) ON DELETE CASCADE ON UPDATE CASCADE
);

DROP TABLE IF EXISTS dbo.REVIEWS;

CREATE TABLE dbo.REVIEWS (
    ReviewID INT PRIMARY KEY,
    Rating INT NOT NULL,
    ReviewText NVARCHAR(1000) NOT NULL,
    ReviewDate DATETIME NOT NULL,
    ClientID INT NULL,
    FreelancerID INT NULL,
    MilestoneID INT NULL,
    CONSTRAINT FK_Reviews_Client FOREIGN KEY (ClientID)
        REFERENCES dbo.CLIENTS(ClientID) ON DELETE NO ACTION ON UPDATE NO ACTION,
    CONSTRAINT FK_Reviews_Freelancer FOREIGN KEY (FreelancerID)
        REFERENCES dbo.FREELANCERS(FreelancerID) ON DELETE NO ACTION ON UPDATE NO
ACTION,
    CONSTRAINT FK_Reviews_Milestone FOREIGN KEY (MilestoneID)
        REFERENCES dbo.MILESTONES(MilestoneID) ON DELETE SET NULL ON UPDATE NO
ACTION
);


DROP TABLE IF EXISTS dbo.TASKS;

CREATE TABLE dbo.TASKS (
    TaskID INT PRIMARY KEY,
    PriorityLevel INT NOT NULL,
    Status NVARCHAR(50) NOT NULL,
    DueDate DATETIME,
    CompletionDate DATETIME,
    TaskDescription NVARCHAR(1000),
    ProjectID INT,
    MilestoneID INT,
    FreelancerID INT,
    CONSTRAINT FK_Tasks_Project FOREIGN KEY (ProjectID)
        REFERENCES dbo.PROJECTS(ProjectID) ON DELETE NO ACTION ON UPDATE NO ACTION,
    CONSTRAINT FK_Tasks_Milestone FOREIGN KEY (MilestoneID)
        REFERENCES dbo.MILESTONES(MilestoneID) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT FK_Tasks_Freelancer FOREIGN KEY (FreelancerID)
        REFERENCES dbo.FREELANCERS(FreelancerID) ON DELETE NO ACTION ON UPDATE NO
ACTION
);

DROP TABLE IF EXISTS dbo.APPLICATIONS;
CREATE TABLE dbo.APPLICATIONS (
    ApplicationID INT PRIMARY KEY,
    Coverletter NVARCHAR(1000) NOT NULL DEFAULT '',
    ExpectedPay DECIMAL(18, 2) NOT NULL DEFAULT 0.00,
    ApplicationDate DATETIME NOT NULL DEFAULT GETDATE(),
    Status NVARCHAR(50) NOT NULL DEFAULT 'Pending',
    ProjectID INT NOT NULL,
    FreelancerID INT NOT NULL,
    CONSTRAINT FK_Applications_Project FOREIGN KEY (ProjectID)
        REFERENCES dbo.PROJECTS(ProjectID)
        ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT FK_Applications_Freelancer FOREIGN KEY (FreelancerID)
        REFERENCES dbo.FREELANCERS(FreelancerID)
        ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT CHK_ExpectedPay CHECK (ExpectedPay >= 0),
    CONSTRAINT CHK_Status CHECK (Status IN ('Pending', 'Accepted', 'Rejected'))
);
```