**DATABASE LA APPEL EventsManagementDB**

-- -- Create Event table

-- CREATE TABLE Event (

-- event\_id SERIAL PRIMARY KEY, -- Auto-incrementing primary key

-- event\_name VARCHAR(255) NOT NULL, -- Event name

-- event\_date DATE NOT NULL, -- Event date

-- location VARCHAR(255) NOT NULL -- Event location

-- );

-- -- Create Budget table

-- CREATE TABLE Budget (

-- budget\_id SERIAL PRIMARY KEY, -- Auto-incrementing primary key

-- event\_id INT NOT NULL, -- Foreign key referencing Event

-- budget\_name VARCHAR(255) NOT NULL, -- Budget name

-- category VARCHAR(255) NOT NULL, -- Budget category

-- allocated\_amount DECIMAL(10, 2) NOT NULL, -- Allocated amount for the budget

-- FOREIGN KEY (event\_id) REFERENCES Event(event\_id) ON DELETE CASCADE

-- );

-- -- Create Expense table

-- CREATE TABLE Expense (

-- expense\_id SERIAL PRIMARY KEY, -- Auto-incrementing primary key

-- budget\_id INT NOT NULL, -- Foreign key referencing Budget

-- expense\_name VARCHAR(255) NOT NULL, -- Expense name

-- category VARCHAR(255) NOT NULL, -- Expense category

-- amount DECIMAL(10, 2) NOT NULL, -- Amount spent on the expense

-- FOREIGN KEY (budget\_id) REFERENCES Budget(budget\_id) ON DELETE CASCADE

-- );

CREATE TABLE Task (

-- task\_id SERIAL PRIMARY KEY,

-- task\_description TEXT NOT NULL,

-- deadline TIMESTAMP,

-- status VARCHAR(50),

-- event\_id INT,

-- CONSTRAINT fk\_event

-- FOREIGN KEY(event\_id)

-- REFERENCES Event(event\_id)

-- ON DELETE CASCADE

-- );

-- -- Create table for Client

-- CREATE TABLE Client (

-- client\_id SERIAL PRIMARY KEY,

-- name VARCHAR(255) NOT NULL,

-- contract\_number VARCHAR(100) UNIQUE

-- );

-- -- Create table for Vendor

-- CREATE TABLE Vendor (

-- vendor\_id SERIAL PRIMARY KEY,

-- vendor\_name VARCHAR(255) NOT NULL,

-- contact\_number VARCHAR(100) UNIQUE

-- );

-- -- Create table for Contract

-- CREATE TABLE Contract (

-- contract\_id SERIAL PRIMARY KEY,

-- contract\_description TEXT NOT NULL,

-- vendor\_id INT,

-- CONSTRAINT fk\_vendor

-- FOREIGN KEY(vendor\_id)

-- REFERENCES Vendor(vendor\_id)

-- ON DELETE CASCADE

-- );

-- -- Create table for Payment

-- CREATE TABLE Payment (

-- payment\_id SERIAL PRIMARY KEY,

-- amount DECIMAL(10, 2) NOT NULL,

-- contract\_id INT,

-- vendor\_id INT,

-- CONSTRAINT fk\_contract

-- FOREIGN KEY(contract\_id)

-- REFERENCES Contract(contract\_id)

-- ON DELETE CASCADE,

-- CONSTRAINT fk\_vendor\_payment

-- FOREIGN KEY(vendor\_id)

-- REFERENCES Vendor(vendor\_id)

-- ON DELETE CASCADE

-- );

-- -- Create table to link Event and Client (many-to-many relationship)

-- CREATE TABLE EventClient (

-- event\_id INT,

-- client\_id INT,

-- PRIMARY KEY (event\_id, client\_id),

-- CONSTRAINT fk\_event\_client

-- FOREIGN KEY(event\_id)

-- REFERENCES Event(event\_id)

-- ON DELETE CASCADE,

-- CONSTRAINT fk\_client\_event

-- FOREIGN KEY(client\_id)

-- REFERENCES Client(client\_id)

-- ON DELETE CASCADE

-- );

-- -- Insert data into the Event table

-- INSERT INTO Event (event\_name, event\_date, location) VALUES

-- ('Annual Company Picnic', '2024-07-15', 'Central Park'),

-- ('Charity Gala', '2024-10-20', 'Grand Ballroom'),

-- ('Product Launch', '2024-12-01', 'Convention Center');

-- -- Insert data into the Budget table

-- INSERT INTO Budget (event\_id, budget\_name, category, allocated\_amount) VALUES

-- (1, 'Picnic Food Budget', 'Catering', 1500.00),

-- (2, 'Gala Venue Budget', 'Venue', 5000.00),

-- (3, 'Launch Marketing Budget', 'Entertainment', 3000.00);

-- -- Insert data into the Expense table

-- -- Ensure the budget\_id corresponds to the entries in the Budget table

-- INSERT INTO Expense (budget\_id, expense\_name, category, amount) VALUES

-- (1, 'Food Catering', 'Catering', 1200.00),

-- (1, 'Beverages', 'Catering', 300.00),

-- (2, 'Venue Booking', 'Venue', 4500.00),

-- (2, 'Decorations', 'Decorations', 500.00),

-- (3, 'Marketing Campaign', 'Entertainment', 2000.00),

-- (3, 'Launch Event', 'Entertainment', 1000.00);